MVG Assignment 1
Elements of Projective Geometry

Instructions

- Due date: April 20 2021.
- Submission link: `https://www.dropbox.com/request/TJm3HXGPmRe0VFv92CpT`
- You should submit a zip file containing: A pdf file with all complete solutions and the code with a main script which calls the required scripts/functions for each computer exercise.
- The report should be written individually, however you are encouraged to work together.
- We highly recommend typing your solutions in Latex.
- Notice that in the end of this exercise you are given a list of useful matlab commands for all the computer exercises.

Points in Homogeneous Coordinates.

Exercise 1.

What is the 2D Cartesian coordinates of the points with homogeneous coordinates:

$$\mathbf{x}_1 = \begin{bmatrix} 4 \\ -2 \\ 2 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 3 \\ -2 \\ -1 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 4\lambda \\ -2\lambda \\ 2\lambda \end{bmatrix}, \lambda \neq 0 \tag{1}$$

What is the interpretation of the point with homogeneous coordinates:

$$\mathbf{x}_4 = \begin{bmatrix} 4 \\ -2 \\ 0 \end{bmatrix} \tag{2}$$

Computer Exercise 1.

Write a matlab function `pflat` that divides the homogeneous coordinates with their last entry for points of any dimensionality. (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.

Lines

Exercise 2.

Compute the homogeneous coordinates of the intersection (in $\mathbb{P}^2$) of the lines, and **find** the corresponding point in $\mathbb{R}^2$.

$$\mathbf{l}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{l}_2 = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \tag{3}$$

Compute the intersection (in $\mathbb{P}^2$) of the lines and **explain** the geometric interpretation in $\mathbb{R}^2$.

$$\mathbf{l}_3 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \mathbf{l}_4 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{4}$$

Figure 1: compEx2.jpg

Compute the line that goes through the points with Cartesian coordinates:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{5}$$

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

Computer Exercise 2.

Load and plot the image in `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 $\mathbf{x} = (x_1, x_2)$ in Cartesian coordinates and a line $\mathbf{l} = (a, b, c)$ can be computed using the distance formula:

$$d = \frac{|ax_1 + bx_2 + c|}{\sqrt{a^2 + b^2}} \tag{6}$$

Compute the distance between the first line and the the intersection point. **Is it** close to zero? Why/why not?

Projective Transformations

Exercise 3.

Let H be the projective transformation:

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{7}$$

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

$$\mathbf{x_1} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{x_2} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

Compute the line $\mathbf{l_1}$ which contains $\mathbf{x_1}, \mathbf{x_2}$, and the line $\mathbf{l_2}$ which contains $\mathbf{y_1}, \mathbf{y_2}$.
Compute $(H^{-1})^T \mathbf{l_1}$ and compare to $\mathbf{l_2}$.
Show that projective transformations preserve lines. That is, for each line $\mathbf{l_1}$ there is a corresponding line $\mathbf{l_2}$ such that if $\mathbf{x}$ belongs to $\mathbf{l}_1$ then the transformation $\mathbf{y} \sim H\mathbf{x}$ belongs to $\mathbf{l}_2$. (Hint: if $\mathbf{l}_1^T\mathbf{x} = 0$ then $\mathbf{l}_1^T H^{-1} H\mathbf{x} = 0$).

Computer Exercise 3.

The file `compEx3.mat` contains the start and end points of a set of lines. Plotting the lines gives the grid in Figure 2. For each of the projective mappings given by the matrices:

$$H_1 = \begin{bmatrix} \sqrt{3} & -1 & 1 \\ 1 & \sqrt{3} & 1 \\ 0 & 0 & 2 \end{bmatrix} H_2 = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$H_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} H_4 = \begin{bmatrix} \sqrt{3} & -1 & 1 \\ 1 & \sqrt{3} & 1 \\ 1/4 & 1/2 & 2 \end{bmatrix} \tag{10}$$

Compute the transformations of the given start and endpoints and plot the lines between them. (Note that you do not need to loop over the points. One matrix multiplication for the start and end points is enough. To compute cartesian coordinates you can use your `pflat` function. Don't forget to use the axis equal command, otherwise the figures might look distorted.)
Which of the transformations preserve lengths between points? Which preserve angles between lines?

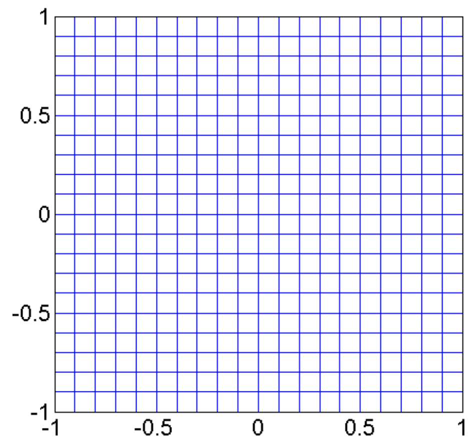

Figure 2: Lines between start and end points

Which maps parallel lines to parallel lines? **Classify** the transformations into euclidean, similarity, affine and projective transformations.

The Pinhole Camera

Exercise 4.

Given a camera matrix $P = \begin{bmatrix} M & \mathbf{p}_4 \end{bmatrix}$ and a 3D point in homogeneous coordinates $\mathbf{x} = (X, Y, Z, 1)^T$ **derive** an expression for the depth of the point, i.e., the projection of the point $\mathbf{x}$ on the principal axis where the projected point to the image plane is given by $w(x, y, 1)^T = P\mathbf{x}$.

Exercise 5.

Compute the projections of the 3D points with homogeneous coordinates

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \tag{11}$$

In the camera with camera matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{12}$$

(a) compEx4im1.jpg

(b) compEx4im2.jpg

Figure 3

What is the geometric interpretation of the projection of $\mathbf{x}_3$?

Computer Exercise 4.

Load and plot the images `compEx4im1.jpg` and `compEx4im2.jpg` (see Figure 3).
The file `compEx4.mat` contains the camera matrices P_1, P_2 and a point model U of the statue.
Compute the camera centers and principal axes of the cameras.
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). In addition **plot** a vector in the direction of the principal axes (viewing direction) from the camera center.
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?

Exercise 6.

Consider the calibrated camera pair $P_1 = \begin{bmatrix} I & \mathbf{0} \end{bmatrix}$ and $P_2 = \begin{bmatrix} R & \mathbf{t} \end{bmatrix}$. If $\mathbf{x}$ is the 2D projection of the 3D point $\mathbf{u}$ in P_1, that is $\mathbf{x} \sim P_1\mathbf{u}$, **verify** that

$$\mathbf{u} \sim \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix} \tag{13}$$

where $s \in \mathbb{R}$. That is, for any s the point of the form $\mathbf{u}(s) = (\mathbf{x}^T, s^T)^T$ 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 $\mathbf{s}$ belongs to the plane

$$\Pi = \begin{bmatrix} \pi \\ 1 \end{bmatrix} \tag{14}$$

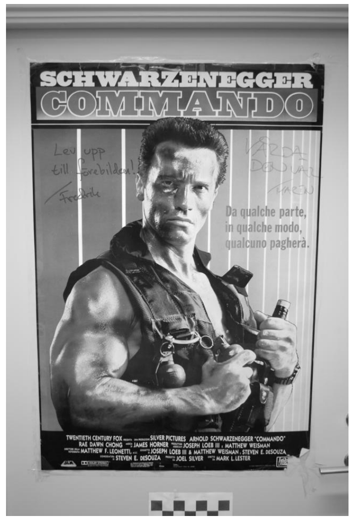

Figure 4: CompEx5.jpg

where $\pi \in \mathbb{R}^3$. Compute 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 $\Pi^T \mathbf{u}$ then the homography

$$H = (R - \mathbf{t}\pi^T) \tag{15}$$

where $P_2 = \begin{bmatrix} R & \mathbf{t} \end{bmatrix}$, maps $\mathbf{x}$ to $\mathbf{y}$. (Hint: What is $P_2\mathbf{u}(s)$ for the s from above?)

Computer Exercise 5.

Figure 4 shows an image (`compEx5.jpg`) of a poster located somewhere in the Centre for Mathematical Sciences. (Here the axes units are pixels.)
The file `compEx5.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

$$P_1 = K \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \tag{16}$$

The goal of this exercise is to create an image of the poster taken by a camera 2m to the right of the poster.
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 6 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. 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 $\begin{bmatrix} I & \mathbf{0} \end{bmatrix}$. Using the results from Exercise 6 **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 right of P_1) and orientation

$$R = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0\frac{1}{2} & \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \tag{17}$$

(minus 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 6 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

$$\tilde{\mathbf{y}} \sim H\tilde{\mathbf{x}} \iff K^{-1}\mathbf{y} \sim HK^{-1}\mathbf{x} \iff \mathbf{y} \sim KHK^{-1}\mathbf{x} \tag{18}$$

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.

Usefull Matlab commands

1

- x(end ,:) % Extracts the last row of x
- repmat(a ,[m n]) % Creates a block matrix with mn copies of a.
- 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 .

2

- 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

3

- plot([startpoints(1 ,:); endpoints(1 ,:)] [startpoints(2 ,:); endpoints(2 ,:)], 'b-'); % Plots a blue line between each startpoint and endpoint

4

- 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) % Plot a vector v starting from the point a, and rescales the by s
- plot(x1(1 ,:) , x1(2 ,:) , '.','Markersize' ,2); % Same as plot but with smaller points

5

- 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 = maketform ('projective ',Htot '); % Creates a projective transformation that can be used in imtransform % NOTE : Matlab uses the transposed version of the homography.

- [new_im ,xdata , ydata] = imtransform(im ,tform ,'size',size(im)); % Creastes a transformed image (using tform) of the same size as the original one.
- imagesc(xdata ,ydata , new_im); % plots the new image with xdata and ydata on the axes