COMPUTER VISION LAB ASSIGNMENT 3
March 12, 2021

Sjoerd Bruin (s2563304) & Jasper van Thuijl (s2503204)

University of Groningen

1. INTRODUCTION

In many computer vision problems, one needs to infer information about a 3D scene from a 2D perspective projection image. In this report we will show how the orientation of a set of two or more parallel 3D lines can be determined from a perspective image of these lines. Moreover, we will show how you can determine the focal length of a camera and the normal of a plane using just two sets of parallel lines.

2. EXERCISE 1

In the 2D and 3D projection of the rectangle there are two sets of parallel lines (the opposite sides are parallel), given by $\vec{w}_1$ and $\vec{w}_2$. In order to find the direction vectors $\vec{w_1}$ we will need to calculate the camera constant f.

First we will determine the two dimensional direction vectors for each set of parallel lines from the 2D image. To do this we have to obtain the start points a, b and the end points c, d for each parallel line i. The obtained endpoints can be used to calculate g_i and h_i as follows:

$$\eta \cdot \begin{pmatrix} g_i \\ h_i \end{pmatrix} = \begin{pmatrix} c - a \\ d - b \end{pmatrix}$$

As shown in 4.2.2, we can use these parameters to construct a $N \times 3$ matrix A, where each row is given by equation 1.

$$A_i = \begin{pmatrix} h_i \cdot f & -g_i \cdot f & b \cdot g_i - a \cdot h_i \end{pmatrix} \tag{1}$$

We can now define a direction vector $\vec{w} = (w_1, w_2, w_3)^T$. The goal is to find a a solution to Equation 2.

$$A\vec{w} = \vec{0} \tag{2}$$

Since this system is over-determined most of the time, there exists infinitely many solutions. We try to obtain the minimum solution that preserves the normalization of $\vec{w}$. We can find the solution by taking the singular value decomposition (SVD), which is defined as $\mathbf{A} = \mathbf{USV}^T$, and extracting the minimal row vector in the matrix $\mathbf{V}$. This vector represents the closest solution to the original $\vec{w}$ that we can find. However, since f is unknown, we will define a vector $\vec{x}$ by:

$$x_1 = w_1 f,\ x_2 = w_2 f,\ x_3 = w_3 \tag{3}$$

Now we can define an A' that is independent of f.

$$A'\vec{x} = \begin{pmatrix} 0 \\ ... \\ 0 \end{pmatrix}$$

Each line of A' is defined by Equation 4.

$$\mathbf{A}'_i = \begin{pmatrix} h_i & -g_i & b \cdot g_i - a \cdot h_i \end{pmatrix} \tag{4}$$

In order to find f we can exploit the perpendicularity property: As both sets of parallel lines are perpendicular to each other, this implicates that $\vec{w}_1 \cdot \vec{w}_2 = 0$.

We can define the results for the first set of parallel lines as (x_1, x_2, x_3) and the results for the second set as (x_4, x_5, x_6). We note the following six definitions:

$$\begin{aligned} w_1 = \frac{x_1}{f}; w_2 = \frac{x_2}{f}; w_3 = x_3 \\ w_4 = \frac{x_4}{f}; w_5 = \frac{x_5}{f}; w_6 = x_6 \end{aligned} \tag{5}$$

Now we can relate the two directional vectors to find the focal length in equation 6:

$$\vec{w}_1 \cdot \vec{w}_2 = w_1 w_4 + w_2 w_5 + w_3 w_6 = 0$$

$$\begin{aligned} \frac{1}{f^2} x_1 x_4 + \frac{1}{f^2} x_2 x_5 + x_3 x_6 = 0 \\ \frac{1}{f^2}(x_1 x_4 + x_2 x_5) = -x_3 x_6 \\ f^2 = -\frac{x_1 x_4 + x_2 x_5}{x_3 x_6} \end{aligned}$$

$$f = \sqrt{-\frac{x_1 x_4 + x_2 x_5}{x_3 x_6}} \tag{6}$$

Equation 6 allows us to determine the focal length f in pixels. The term under the square root must be non-negative, of course, but we will assume that this is the case in this theoretical exercise. Now that f is calculated, we can use Equation 3 again to obtain the direction vectors $\vec{w}$ for each of the sides.

3. EXERCISE 2

In this exercise, we were requested to process a picture taken by a camera, containing a number of concentric rectangles with parallel sides. This input image can be observed in Figure 1. The provided `scanlines` function can be easily used to determine the endpoints of the parallel lines using the mouse. Two data sets of endpoints were selected. All lines from the left side of the center were selected. Afterwards the resulting x and y arrays were saved using the *savescanpoints* function in `par_lines1.dat`. Similarly, all lines from the top side of the center were selected and saved in `par_lines2.dat`. The saved files are structured as follows. Each selected line is entered on a new line, and has 4 columns. The first two contian the startpoint x_1, y_1, the last two the endpoint x_2, y_2 (referred to as a, b, c, d in Exercise 1). These files are will be used in Exercise 4.

4. EXERCISE 3

In this exercise, we were asked to study the file `par_line.m` and to make sure we understood the contents. We will explain how this file works in this section.

The first thing that is done is to load a .dat file and extract an $N \times 4$ array, where each row i contains the definition of a line. The line definition is formatted as follows: the first two numbers a, b give the start of the line, while the latter two numbers c, d give the end of the line. We can use these to determine the two-dimensional direction vector:
$\eta \cdot \begin{pmatrix} g_i \\ h_i \end{pmatrix} = \begin{pmatrix} c-a \\ d-b \end{pmatrix}$.

The function then builds an $N \times 3$ matrix $\mathbf{A}$, where each row is given by equation 7.

$$A_i = \begin{pmatrix} h_i \cdot f & -g_i \cdot f & b \cdot g_i - a \cdot h_i \end{pmatrix} \tag{7}$$

Next, we define a vector $\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$. Our goal is to find the solution to equation 8:

$$\mathbf{A}\vec{w} = \vec{0} \tag{8}$$

Finding the solution to this equation is difficult because the problem is overdetermined most of the time, and therefore there exist infinitely many solutions. We are interested in the minimum solution that preserves the normalisation of $\vec{w}$. To do this, we instead minimise the quantity $||\mathbf{A}\vec{w}||^2$ with the additional constraint that $||\vec{w}|| = 1$. We can find the solution by taking the singular value decomposition (SVD), which is defined as $\mathbf{A} = \mathbf{USV}^T$, and extracting the minimal row vector in the matrix $\mathbf{V}$. This vector represents the closest solution to the original $\vec{w}$ that we can find.

There is one complication: the above method only works if the value of f is known in advance. In our case, we cannot assume that this is the case. To make our process independent of f, we redefine the matrix $\mathbf{A}$ as $\mathbf{A}'$, with row elements as given in equation 9.

$$\mathbf{A}'_i = \begin{pmatrix} h_i & -g_i & b \cdot g_i - a \cdot h_i \end{pmatrix} \tag{9}$$

The definition of this matrix is done in lines 11-14 in `par_line.m`. With the matrix given in equation 9, we can now solve for the vector $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ using SVD as discussed above, with $\vec{x}$ substituted for $\vec{w}$. The SVD computation and the selection of the minimal vector from $\mathbf{V}$ is done in lines 16-18 of `par_line.m`. We can compute the original direction vector $\vec{w}$ once we know the true value of the focal length, using $w_1 = \frac{x_1}{f}$, $w_2 = \frac{x_2}{f}$, and $w_3 = x_3$. These values of $\vec{w}$ still need to be normalised after this operation.

5. EXERCISE 4

The major problem in the approach described in exercise 3 is that we do not know the focal length, and thus we cannot find the vector $\vec{w}$. However, if we have some additional information, we might be able to use that to add an extra constraint to the problem, which subsequently allows us to determine the value of f. In this exercise, we analyse a rectangle, which has two sets of parallel lines, given by $\vec{w}_1$ and $\vec{w}_2$, which are perpendicular to one another, meaning that $\vec{w}_1 \cdot \vec{w}_2 = 0$. We can use this to our advantage by noting from exercise 3 that f appears in the definition of x_1 and x_2. We can define such relationships for both sets of parallel lines. By convention, we denote the results for the one set of parallel lines as (x_1, x_2, x_3) and the results for the other set as (x_4, x_5, x_6). First of all, we note the following six definitions:

$$\begin{aligned} w_1 = \frac{x_1}{f}; w_2 = \frac{x_2}{f}; w_3 = x_3 \\ w_4 = \frac{x_4}{f}; w_5 = \frac{x_5}{f}; w_6 = x_6 \end{aligned} \tag{10}$$

Second of all, we can relate the two directional vectors to find the focal length in equation 11:

$$\vec{w}_1 \cdot \vec{w}_2 = w_1 w_4 + w_2 w_5 + w_3 w_6 = 0$$

$$\begin{aligned} \frac{1}{f^2} x_1 x_4 + \frac{1}{f^2} x_2 x_5 + x_3 x_6 = 0 \\ \frac{1}{f^2}(x_1 x_4 + x_2 x_5) = -x_3 x_6 \\ f^2 = -\frac{x_1 x_4 + x_2 x_5}{x_3 x_6} \\ f = \sqrt{-\frac{x_1 x_4 + x_2 x_5}{x_3 x_6}} \end{aligned} \tag{11}$$

Equation 11 allows us to determine the focal length f in pixels (not in millimeters!). The term under the square root must be non-negative, of course, but in the example for this exercise, this has proven to be the case.

Now that we know how to find the focal length, we can apply the definitions in equation 10 to find the w_i direction vector elements from the x_i elements. After computing this, we must normalise the direction vectors.

The last thing we were asked to do was to determine the normal of the plane. The normal of the plane is given by the vector $\vec{n} = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$, where A, B, and C are elements of the plane equation. Because the three elements of the normal vectors are unknown, we need to produce at least three equations which involve only those variables as unknowns. We can produce these equations using either equation 12 or 13, which hold because any vector lying in the plane must be perpendicular to the normal vector.

$$\vec{n} \cdot \vec{w} = Aw_1 + Bw_2 + Cw_3 = 0 \tag{12}$$

$$\vec{n} \cdot \vec{w} = Au_\infty + Bv_\infty + f = 0 \tag{13}$$

The vanishing points in equation 13 can be computed using the definitions in equation 14.

$$\begin{aligned} u_{inf} &= f\frac{w_1}{w_3} \\ v_{inf} &= f\frac{w_2}{w_3} \end{aligned} \tag{14}$$

We can produce a total of four simultaneous equations from equations 12 and 13, two for each equation by inserting one of the two directional vectors. In this exercise, we use the following matrix M to solve the equation:

$$M = \begin{pmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ u_{inf}^{(1)} & v_{inf}^{(1)} & f \\ u_{inf}^{(2)} & v_{inf}^{(2)} & f \end{pmatrix} \tag{15}$$

With matrix M, we can using singular value decomposition once again to solve for the minimal vector in matrix V. We have to solve the following equation:

$$M\vec{n} = \vec{0} \tag{16}$$

The discussion above lays out the implementation of our program. We reuse code from the `par_line.m` matlab file that was provided to us. Specifically, we use that code to load in the parallel lines data sets, and to determine the value of the vectors $\vec{x}$. After that, we use equation 11 to find f. With f, we compute the correct values of $\vec{w}$, and normalise the result such that the norm of $\vec{w}$ is equal to 1. Then, we compute the vanishing points using equation 14, construct matrix M using equation 15, and solve equation 16 using singular value decomposition. The results of the above operations give us a value for f in pixels, two direction vectors for each set of parallel lines, and a normal vector for the plane. The parallel lines that we drew are included in the zip file that we have handed in. The results of applying the above operations to the two sets of parallel lines are given below.

$f = 1187.9$

$\vec{w}_1 = \begin{pmatrix} -0.6015 \\ 0.5728 \\ 0.5568 \end{pmatrix}$

$\vec{w}_2 = \begin{pmatrix} 0.7973 \\ 0.3865 \\ 0.4637 \end{pmatrix}$

$\vec{n} = \begin{pmatrix} -0.0503 \\ -0.7229 \\ 0.6892 \end{pmatrix}$

It should be noted that the result of f is in pixels, and not in millimeters, so the value is difficult to relate to the lens' focal length (which would be given in mm).

In order to understand the results of the assignment, we need to consider the rectangle as it is shown in figure 1. We need to construct the coordinate system as the programme uses it. The xy plane is defined as the image axes. The z-axis points into the image.

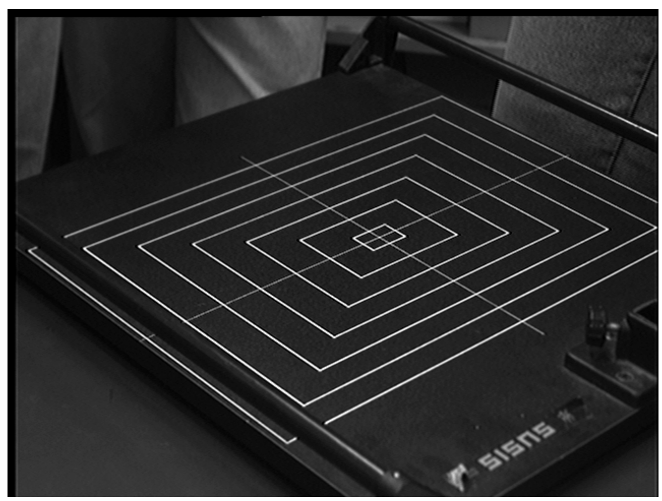

Fig. 1: Rectangle used for inferring parallel lines.

The two direction vectors $\vec{w}_1$ and $\vec{w}_2$ give the direction in 3D space, which we can only approximately verify in the scope of this assignment (that is, without doing much more complicated computations). We see that $\vec{w}_1$ has a somewhat large negative x value, which indicates that the lines move in a direction that tends to the left. Furthermore, the lines have a rather steep ascent in the y-direction. This of course makes sense for the parallel lines the move from lower-right to upper-left. For $\vec{w}_2$, it becomes clear that there is a large velocity in the positive x-direction, meaning that the lines move quite horizontally to the right. The y-direction is positive but less steep than for $\vec{w}_1$. These characteristics make sense for the lines that move from lower-left to upper-right. For the z-direction, we intuit that the lines associated with $\vec{w}_1$ move in the positive z-direction (away from the image plane) and the lines associated with $\vec{w}_2$ do the same. Indeed, both direction vectors have a positive z value, agreeing with this intuition.

We did perform a sanity check on these two vectors by computing the dot product between them. After all, the two vectors should be perpendicular, and therefore the dot product should be zero (or as close to zero as is numerically possible). The result turned out to be $\vec{w}_1 \cdot \vec{w}_2 = -1.1102 \cdot 10^{-16}$, which is an error on the order of 10^{-16}, which is quite negligible.

It becomes clear that the normal points away from us in the z-direction (a positive z-value), the y-direction (bottom-top) points down (a negative y-value) and the x-direction (left-right) points very slightly to the left (a small negative x-value). This is not what we would expect when looking at the surface of the rectangle. Indeed, it is the opposite of what we expect. The reason for this is simple: our mathematical operation for defining simultaneous equations, using equations 12 and 13, only enforces the requirement that the normal is perpendicular to the direction vectors. This has two possible solutions, one representing the back of the rectangular plane, and one representing the front of the rectangular plane (as seen in the image). In this case, the programme has found the normal vector associated with the back of the rectangular plane. We can solve this by negating the normal vector, which gives the following result:

$$\vec{n} = \begin{pmatrix} 0.0503 \\ 0.7229 \\ -0.6892 \end{pmatrix}$$

We can enforce that the normal is always forward-facing (towards the camera) by noting that the normal should point in the direction of the negative z-axis, so that its z component points towards the image. Indeed, the negative z-direction can be considered as the view direction, and the normal should point in such a direction that we can see the surface it represents (so not the back of the surface). To ensure this, we simply need to check whether the z component is positive. If it is, then we negate the normal so that we get the forward-facing normal instead.

We encountered this issue in the final stages of writing this report, and thus we decided to include the discussion, as we believe it is valuable for understanding what might go wrong when doing such experiments. In the final version of our code, we have included the z-direction check, which then flips the normal to its correct orientation. You will therefore not get the backward-facing normal when running the programme.

One last sanity check that we can perform is to make sure that the normal is perpendicular to $\vec{w}_1$ and $\vec{w}_2$ by computing the dot product. We find that $\vec{n} \cdot \vec{w}_1 = 1.2143 \cdot 10^{-16}$ and $\vec{n} \cdot \vec{w}_2 = 2.0817 \cdot 10^{-17}$, both of which are so negligibly close to zero that we can ascribe the difference to numerical imprecision and to minor sampling errors when constructing the parallel lines in exercise 2.

To summarise, we have found the focal length, the directions of two sets of parallel lines, and the normal vector to the plane of the rectangle. We believe that the results we found make sense, and the mathematical checks we could perform on them confirm that the results are plausible.

6. CONCLUSIONS

1. We found that we could determine the direction of parallel lines without knowing the focal length beforehand simply by using two mutally perpendicular sets of parallel lines, for example from a set of rectangles, and using the perpendicularity property to solve for the focal length. This then allowed us to find the direction vectors and even the normal of the plane of the rectangle that we used to define the parallel lines.