4 2D → 3D inference

In this session the topic is 3D Scene Analysis. We start by explaining how images of 3D scenes are made by perspective projection. Then we look at a very simple 2D→ 3D inference problem.

4.1 Central or perspective projection

In central projection (**cp**) we have a distinguished point called the *center of projection* C and a plane, called the *image plane*, where the image will be formed. Each 3D point $P = (x, y, z)$ is projected upon the image plane by connecting P and C by a straight line. The intersection point of this line with the image plane is called the *projection* of P.

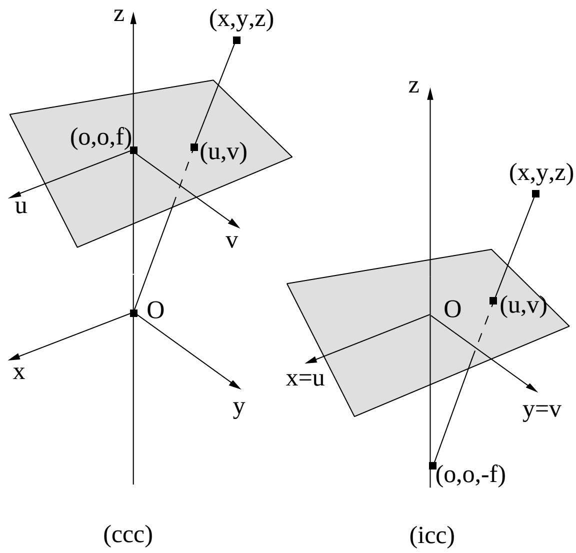

Figure 3: *Central projection using the pinhole-camera model. Left: camera-centered (***ccc***). Right: image centered (***icc***).*

This type of projection occurs in two variants:

- *camera-centered coordinate system* (**ccc**): Here we put the camera in the origin of the coordinate system with the image plane parallel to the x-y plane at a distance f (this is the *camera constant*, usually identified with the *focal length* as explained above), see Fig. 3. So the projection center $C = (0, 0, 0)$. The z-axis is called the *optical axis*. If we use the variables u and v as coordinates in the image plane, then the projection can be described by the equation:

$$\boxed{u = fx/z, \quad v = fy/z.} \tag{4.1}$$

  It is sensible to use these equations only for points *above* the image plane, that is, for $z > f$.

- *image-centered coordinate system* (**icc**): Here the projection center $C = (0, 0, -f)$, so we put the camera at a distance f below the image plane, which is the x-y plane in this case, see Fig. 3. This projection can be described by the equation:

$$\boxed{u = fx/(f + z), \quad v = fy/(f + z).} \tag{4.2}$$

  Again, it is sensible to use these equations only for $z > 0$.

4.1.1 Parallel projection

As an approximation to central projection *parallel projection* (**pp**) is obtained: take the limit $f \to \infty$ in the **icc** case to find

$$u = x \quad v = y.$$

These equations say that the projection takes place along straight lines perpendicular to the projection plane. This approximation is valid if the objects are very small compared to f or if they are very far removed from the camera.

The basic question of 3D vision now is to what extent 3D properties can be reconstructed from the projective images by inverting the projection process, that is, by *backprojection*.

4.1.2 Lines to lines

Under perspective projection a 3D line is mapped onto a 2D line. This can be proved as follows. Assume a **ccc** projection, see (4.1). Let $\vec{p}, \vec{w} \in \mathbb{R}^3$ and let L be the 3D line with representation:

$$L = \left\{ \vec{p} + \lambda\vec{w} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} + \lambda \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} : \lambda \in \mathbb{R} \right\}. \tag{4.3}$$

The vector $\vec{w}$ is called the 'direction vector' and will be assumed to be normalized: $\|\vec{w}\| = 1$. Then the coordinates (u, v) of the projection of this line have the form:

$$\begin{array}{rcl} u & = & f(p_1 + \lambda w_1)/(p_3 + \lambda w_3) \\ v & = & f(p_2 + \lambda w_2)/(p_3 + \lambda w_3) \end{array} \tag{4.4}$$

Now a little algebra shows that

$$\begin{pmatrix} u \\ v \end{pmatrix} = f \begin{pmatrix} p_1/p_3 \\ p_2/p_3 \end{pmatrix} + \mu \begin{pmatrix} w_1 - (p_1/p_3)w_3 \\ w_2 - (p_2/p_3)w_3 \end{pmatrix}, \qquad \mu = \frac{f\lambda}{p_3 + \lambda w_3},$$

which is the parameter representation of a 2D line (when λ runs over $\mathbb{R}$ so does μ).

4.1.3 Relation between line parameters

There is a relation between the parameters of a 3D line and those of its perspective projection. Represent the 3D line L as

$$L = \left\{ \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} + \lambda \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} : \lambda \in \mathbb{R} \right\}$$

and the perspective projection of this line (again a line) as

$$L' = \left\{ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \eta \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} : \eta \in \mathbb{R} \right\}$$

Taking a **ccc** system, we find by (4.1) that λ, η must satisfy

$$c_1 + \eta d_1 = f(p_1 + \lambda w_1)/(p_3 + \lambda w_3) \tag{4.5}$$

$$c_2 + \eta d_2 = f(p_2 + \lambda w_2)/(p_3 + \lambda w_3) \tag{4.6}$$

Eliminating η we find

$$(c_1 d_2 - c_2 d_1)p_3 - (d_2 p_1 - d_1 p_2)f + \lambda[(c_1 d_2 - c_2 d_1)w_3 - (d_2 w_1 - d_1 w_2)f] = 0.$$

Now this equation must be true for any λ. Therefore the following two equations must hold:

$$\boxed{\begin{array}{rcl} d_2 f p_1 - d_1 f p_2 + (c_2 d_1 - c_1 d_2)p_3 & = & 0 \\ d_2 f w_1 - d_1 f w_2 + (c_2 d_1 - c_1 d_2)w_3 & = & 0 \end{array}} \tag{4.7}$$

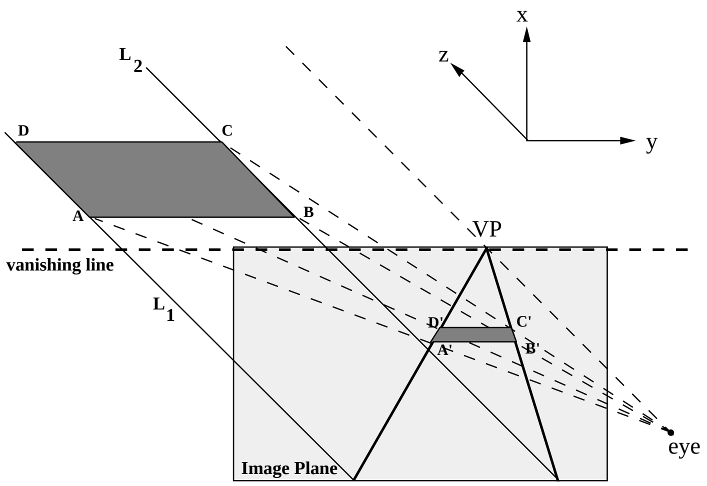

Figure 4: *Vanishing point VP of parallel 3D lines L_1 and L_2 in a plane parallel to the y-z plane. The eye is at the projection center. Dark regions: square ABCD with projection A'B'C'D'.*

4.1.4 Vanishing point

Perspective projection of parallel 3D lines having nonzero slope along the z-axis meet (in general) in a *vanishing point* on the image projection plane, see Fig. 4. This point can be found as the intersection with the image plane of a line through the projection center parallel to the given 3D lines. This can be seen as follows. Consider again the 3D line L with representation (4.3) and projection (4.4). For lines with nonzero slope along the z-axis, $w_3 \neq 0$. Points on the line which are infinitely far removed from the projection center will have coordinates u_∞, v_∞ given by

$$\begin{aligned} u_\infty &= \lim_{\lambda\to\infty} f(p_1+\lambda w_1)/(p_3+\lambda w_3) = f(w_1/w_3) \\ v_\infty &= \lim_{\lambda\to\infty} f(p_2+\lambda w_2)/(p_3+\lambda w_3) = f(w_2/w_3) \end{aligned} \tag{4.8}$$

Now the point (u_∞, v_∞) is independent of p_1, p_2, p_3, so all lines with the same direction vector $\vec{w} = (w_1, w_2, w_3)$ — which are therefore *parallel* — with nonzero slope along the z-axis have perspective projections which meet in a single point, called the *vanishing point*.

Lines L with $w_3 = 0$ have perspective projections which are parallel to L. For, in this case

$$\begin{pmatrix} u \\ v \end{pmatrix} = f \begin{pmatrix} p_1/p_3 \\ p_2/p_3 \end{pmatrix} + \mu \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \qquad \mu = \frac{f\lambda}{p_3},$$

which is the parameter representation of a 2D line with direction vector proportional to $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$. It is therefore parallel to L which has direction vector $\begin{pmatrix} w_1 \\ w_2 \\ 0 \end{pmatrix}$.

4.1.5 Vanishing line

We now show that the vanishing points of all 3D lines which lie in parallel planes all lie on a line in the image plane called the *vanishing line*. This line can be found as the intersection of the image plane with the plane through the projection center parallel to the given 3D planes. This can be seen as follows. Consider a plane V with normal (A, B, C). Points in this plane satisfy the equation

$$Ax + By + Cz + D = 0$$

for some D. Let L be a 3D line with dv $\vec{w} = (w_1, w_2, w_3)$ ($w_3 \neq 0$) lying in a plane parallel to V. Then the dv $\vec{w}$ of L must be perpendicular to (A, B, C), in other words:

$$Aw_1 + Bw_2 + Cw_3 = 0.$$

Using (4.8) this can be written as

$$A\frac{u_\infty w_3}{f} + B\frac{v_\infty w_3}{f} + Cw_3 = 0.$$

Multiplying by f and dividing by w_3 (allowed since $w_3 \neq 0$ by assumption) yields

$$Au_\infty + Bv_\infty + Cf = 0.$$

For fixed A, B, C, f this is the equation of a line in the image plane, called the *vanishing line*.

In conclusion, the vanishing points of lines lying in parallel planes with normal vector (A, B, C) lie on the line L_∞ defined by:

$$\boxed{L_\infty = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} : Au + Bv + Cf = 0. \right\}} \tag{4.9}$$

Also, every point on the line L_∞ is the vanishing point of some line in such a plane. This is illustrated in Fig. 5.

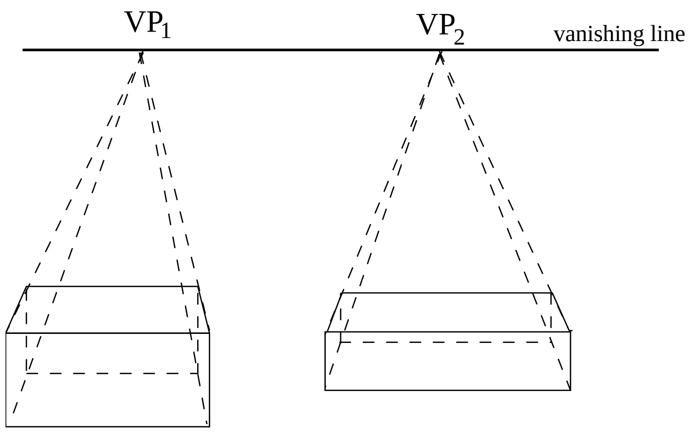

Figure 5: *Lines in parallel planes have vanishing points on the vanishing line.*

4.2 2D→ 3D inference: Direction of 3D parallel lines

Let us show how to determine the orientation of a set of two or more parallel 3D lines from a perspective image of these lines. In particular, the position of the vanishing point will allow us to recover the *direction vector* of the parallel lines. We start with the case of two parallel lines. To reduce noise influences we might want to take measurements of the projections of more than two parallel lines. This case is considered too.

4.2.1 Two parallel lines

Consider two parallel 3D lines with dv $\vec{w}$ whose projections meet in the vanishing point with coordinates u_∞, v_∞ given by (4.8). From this equation and the normalization condition $w_1^2 + w_2^2 + w_3^2 = 1$ we can solve for w_1, w_2, w_3:

$$\boxed{\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \frac{1}{\sqrt{u_\infty^2 + v_\infty^2 + f^2}} \begin{pmatrix} u_\infty \\ v_\infty \\ f \end{pmatrix}.}$$

4.2.2 N parallel lines

Assume that the perspective projection of N parallel lines with unknown direction vector $\vec{w} = (w_1, w_2, w_3)$ is observed. Represent the projection of the nth line by

$$L_n = \left\{ \begin{pmatrix} c_n \\ d_n \end{pmatrix} + \eta \begin{pmatrix} g_n \\ h_n \end{pmatrix} : \eta \in \mathbb{R} \right\}, \quad n = 1, \ldots, N.$$

By the 2D-3D relation (4.7) we have

$$h_n f w_1 - g_n f w_2 + (d_n g_n - c_n h_n) w_3 = 0, \quad n = 1, \ldots, N.$$

In matrix form this can be written as

$$\mathbf{A}\vec{w} = \vec{0},$$

or, explicitly,

$$\boxed{\mathbf{A}\,\vec{w} = \begin{pmatrix} h_1 f & -g_1 f & d_1 g_1 - c_1 h_1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ h_N f & -g_N f & d_N g_N - c_N h_N \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix}} \tag{4.10}$$

The additional constraint is that $\vec{w}$ is a unit vector:

$$\|\vec{w}\| = 1. \tag{4.11}$$

For $N > 2$ this is an overdetermined system of equations (more equations than unknowns). To solve it we use *least squares* techniques, see Appendix A (in particular Section A.6). Since in general the equation $\mathbf{A}\vec{w} = \vec{0}$ has no nontrivial solution satisfying (4.11) we take as solution the vector $\vec{w}$ that minimizes $\|\mathbf{A}\vec{w}\|^2$ and satisfies (4.11). This vector can be computed by performing a *singular value decomposition* of the matrix $\mathbf{A}$ and choosing the right singular vector of $\mathbf{A}$ having smallest singular value.

Note that this method requires that the camera constant f is known.

If we define the vector $\vec{x}$ by

$$x_1 = w_1 f, \quad x_2 = w_2 f, \quad x_3 = w_3 \tag{4.12}$$

we can also solve the equation $\mathbf{A}'\vec{x} = 0$ with $\mathbf{A}'$ *independent of* f:

$$\mathbf{A}'\,\vec{x} = \begin{pmatrix} h_1 & -g_1 & d_1 g_1 - c_1 h_1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ h_N & -g_N & d_N g_N - c_N h_N \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix}. \tag{4.13}$$

Afterwards we then have to compute $\vec{w}$ by

$$w_1 = x_1/f, \quad w_2 = x_2/f, \quad w_3 = x_3 \tag{4.14}$$

followed by a normalization of the vector $\vec{w}$ thus obtained.

4.3 EXERCISES

All the files needed for this practical are found in the archive "inference.zip" provided on Nestor. Download this and type

```
unzip inference.zip
```

in your work directory, followed by

```
cd inference
```

The problem is to determine the orientation of a set of two or more parallel 3D lines from a perspective image of these lines. Now assume that both the direction vector $\vec{w}$ of the parallel lines, and the camera constant f is unknown, but that the available information is extended by giving the perspective projection of a rectangle (or a number of rectangles with parallel sides), see Fig. 6.

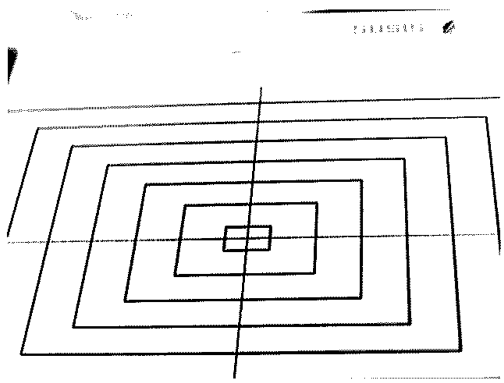

Figure 6: *Camera image of a set of rectangles.*

Exercise 1.

Show how the camera constant f can be obtained from the perspective projection of a rectangle, and from this, the direction vectors of the sides of the rectangles (and therefore the normal of the planar patch containing the rectangles).

Exercise 2.

You will need to process a picture taken by camera for you, containing a number of concentric rectangles with parallel sides as shown in Fig. 6. From the convergence of the parallel lines in the image (both for the horizontal and vertical sides of the rectangles), the vanishing points can be easily determined. The image to be used is in TIFF file format, with filename `rectangle.tif`.

Start MatLab and read in the 'rectangle' image. Now you have to obtain the endpoints of two sets of parallel lines in this image, one set each for the horizontal and vertical sides of the rectangles. To this end a MatLab function plugin `scanpoints` is provided. After reading the image, start scanpoints by typing

```
[x,y] = scanpoints(I);
```

In which `I` is the image. Scan the sides of the rectangles by clicking the left mouse button over their endpoints. Click the right mouse button to end the scanning.

You should see the coordinates of points you clicked on are in variable x and y respectively.

IMPORTANT: scanpoints allows you to scan a single set of parallel lines! Save them in a file called, e.g. `par_lines.dat`. You can save your results in the correct format by calling:

```
savescanpoints('par_lines.dat', x, y);
```

Note that this overwrites any previous file you wrote with the same name. Create two files: one for each set of parallel lines.

Exercise 3.

Look at the file `par_line.m`. This matlab determines the orientation of N parallel 3D lines in a perspective image. It reads in a data file such as `par_lines.dat` (made with the MatLab plugin `savescanpoints`) containing the coordinates of the endpoints of the images of the parallel lines (take a look at the contents), and then computes the least squares solution of the matrix system (4.10). Study the code and try to understand what it does.

Exercise 4.

Modify the routine `par_line.m` to a routine `rectangle.m` which reads two files (e.g. `parlines1.dat` and `parlines2.dat`) and writes the following information:

1. the camera constant f
2. the direction vectors of the sides of the rectangles
3. the normal of the planar patch containing the rectangles

Make sure that you can later during the course retrieve the value of f since you will need this information in the first lab session on Optic flow (or keep the file `parlines1.dat` and `parlines2.dat` so that you can redo the computation).