CHAPTER 1

Cameras

PROBLEMS

1.1. Derive the perspective equation projections for a virtual image located at a distance f' *in front* of the pinhole.

Solution We write again $\overrightarrow{OP'} = \lambda\overrightarrow{OP}$ but this time impose $z' = -f'$ (since the image plane is in front of the pinhole and has therefore negative depth). The perspective projection equations become

$$\begin{cases} x' = -f' \dfrac{x}{z}, \\ y' = -f' \dfrac{y}{z}. \end{cases}$$

Note that the magnification is positive in this case since z is always negative.

1.2. Prove geometrically that the projections of two parallel lines lying in some plane Π appear to converge on a horizon line H formed by the intersection of the image plane with the plane parallel to Π and passing through the pinhole.

Solution Let us consider two parallel lines Δ_1 and Δ_2 lying in the plane Π and define Δ_0 as the line passing through the pinhole that is parallel to Δ_1 and Δ_2. The lines Δ_0 and Δ_1 define a plane Π_1, and the lines Δ_0 and Δ_2 define a second plane Π_2. Clearly, Δ_1 and Δ_2 project onto the lines δ_1 and δ_2 where Π_1 and Π_2 intersect the image plane Π'. These two lines intersect at the point p_0 where Δ_0 intersects Π'. This point is the *vanishing point* associated with the family of lines parallel to Δ_0, and the projection of *any* line in the family appears to converge on it. (This is true even for lines parallel to Δ_0 that do not lie in Π.)
Now let us consider two other parallel lines Δ_1' and Δ_2' in Π and define as before the corresponding line Δ_0' and vanishing point p_0'. The lines Δ_0 and Δ_0' line in a plane parallel to Π that intersects the image plane along a line H passing through p_0 and p_0'. This is the horizon line, and any two parallel lines in Π appears to intersect on it. They appear to converge there since any image point above the horizon is associated with a ray issued from the pinhole and pointing *away* from Π. Horizon points correspond to rays parallel to Π and points in that plane located at an infinite distance from the pinhole.

1.3. Prove the same result algebraically using the perspective projection Eq. (1.1). You can assume for simplicity that the plane Π is orthogonal to the image plane.

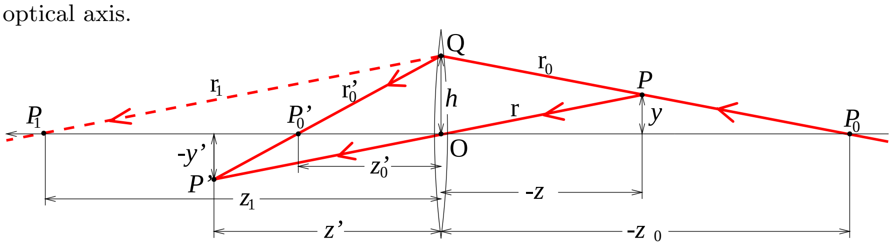
Solution Let us define the plane Π by $y = c$ and consider a line Δ in this plane with equation $ax + bz = d$. According to Eq. (1.1), a point on this line projects onto the image point defined by

$$\begin{cases} x' = f' \dfrac{x}{z} = f' \dfrac{d - bz}{az}, \\ y' = f' \dfrac{y}{z} = f' \dfrac{c}{z}. \end{cases}$$

This is a parametric representation of the image δ of the line Δ with z as the parameter. This image is in fact only a half-line since when $z \to -\infty$, it stops at the point $(x', y') = (-f'b/a, 0)$ on the x' axis of the image plane. This is the vanishing point associated with all parallel lines with slope $-b/a$ in the plane Π. All vanishing points lie on the x' axis, which is the horizon line in this case.

1.4. Derive the thin lens equation.
Hint: consider a ray r_0 passing through the point P and construct the rays r_1 and r_2 obtained respectively by the refraction of r_0 by the right boundary of the lens and the refraction of r_1 by its left boundary.

Solution Consider a point P located at (negative) depth z and distance y from the optical axis, and let r_0 denote a ray passing through P and intersecting the optical axis in P_0 at (negative) depth z_0 and the lens in Q at a distance h from the optical axis.



Before constructing the image of P, let us first determine the image P_0' of P_0 on the optical axis: after refraction at the right circular boundary of the lens, r_0 is transformed into a new ray r_1 intersecting the optical axis at the point P_1 whose depth z_1 verifies, according to (1.5),

$$\frac{1}{-z_0} + \frac{n}{z_1} = \frac{n-1}{R}.$$

The ray r_1 is immediately refracted at the left boundary of the lens, yielding a new ray r_0' that intersects the optical axis in P_0'. The paraxial refraction equation can be rewritten in this case as

$$\frac{n}{-z_1} + \frac{1}{z_0'} = \frac{1-n}{-R},$$

and adding these two equation yields:

$$\frac{1}{z_0'} - \frac{1}{z_0} = \frac{1}{f}, \quad \text{where} \quad f = \frac{R}{2(n-1)}. \tag{1.1}$$

Let r denote the ray passing through P and the center O of the lens, and let P' denote the intersection of r and r_0, located at depth z' and at a distance $-y'$ of the optical axis. We have the following relations among the sides of similar triangles:

$$\left\{\begin{array}{l} \dfrac{y}{h} = \dfrac{z - z_0}{-z_0} = (1 - \dfrac{z}{z_0}), \\ \dfrac{-y'}{h} = \dfrac{z' - z_0'}{z_0'} = -(1 - \dfrac{z'}{z_0'}), \\ \dfrac{y'}{z'} = \dfrac{y}{z}. \end{array}\right. \tag{1.2}$$

Combining Eqs. (1.2) and (1.1) to eliminate h, y, and y' finally yields

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}.$$

1.5. Consider a camera equipped with a thin lens, with its image plane at position z' and the plane of scene points in focus at position z. Now suppose that the image plane is moved to $\hat{z}'$. Show that the diameter of the corresponding blur circle is

$$d\frac{|z' - \hat{z}'|}{z'},$$

where d is the lens diameter. Use this result to show that the depth of field (i.e., the distance between the near and far planes that will keep the diameter of the blur circles below some threshold ε) is given by

$$D = 2\varepsilon f z(z + f)\frac{d}{f^2d^2 - \varepsilon^2 z^2},$$

and conclude that, for a *fixed* focal length, the depth of field increases as the lens diameter decreases, and thus the f number increases.
Hint: Solve for the depth $\hat{z}$ of a point whose image is focused on the image plane at position $\hat{z}'$, considering both the case where $\hat{z}'$ is larger than z' and the case where it is smaller.

Solution If ε denotes the diameter of the blur circle, using similar triangles immediately shows that

$$\varepsilon = d\frac{|z' - \hat{z}'|}{z'}.$$

Now let us assume that $z' > \hat{z}'$. Using the thin lens equation to solve for the depth $\hat{z}$ of a point focused on the plane $\hat{z}'$ yields

$$\hat{z} = fz\frac{d - \varepsilon}{df + \varepsilon z}.$$

By the same token, taking $z' < \hat{z}'$ yields

$$\hat{z} = fz\frac{d + \varepsilon}{df - \varepsilon z}.$$

Finally taking D to be the difference of these two depths yields

$$D = \frac{2\varepsilon df z(z + f)}{f^2d^2 - \varepsilon^2 z^2}.$$

Now we can write $D = kd/(f^2d^2 - \varepsilon^2 z^2)$, where $k > 0$ since $z' = fz/(f + z) > 0$. Differentiating D with respect to d for a fixed depth z and focal length f yields

$$\frac{\partial D}{\partial d} = -k\frac{f^2d^2 + \varepsilon^2 z^2}{(f^2d^2 - \varepsilon^2 z^2)^2} < 0,$$

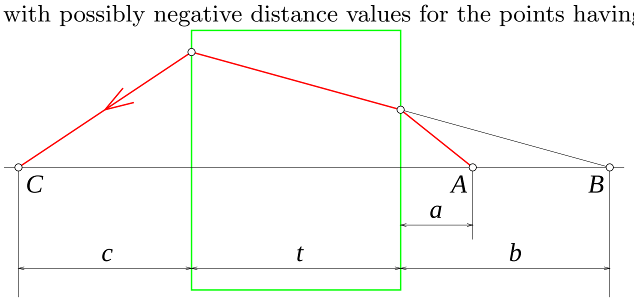
which immediately shows that D decreases when d increases, or equivalently, that D increases with the f number of the lens.

1.6. Give a geometric construction of the image P' of a point P given the two focal points F and F' of a thin lens.

Solution Let us assume that the point P is off the optical axis of the lens. Draw the ray r passing through P and F. After being refracted by the lens, r emerges parallel to the optical axis. Now draw the ray r' passing through P and parallel to the optical axis. After being refracted by the lens, r' must pass through F'. Draw the two refracted rays. They intersect at the image P' of P. For a point P on the optical axis, just construct the image of a point off-axis with the same depth to determine the depth of the image of P'. It is easy to derive the thin lens equation from this geometric construction.

1.7. Derive the thick lens equations in the case where both spherical boundaries of the lens have the same radius.

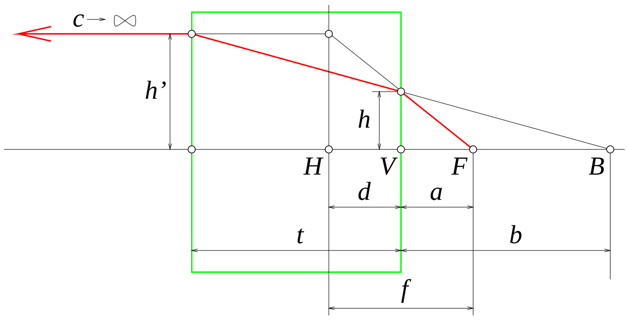
Solution The diagram below will help set up the notation. The thickness of the lens is denoted by t. All distances here are taken positive; if some of the points changed side in a different setting, all formulas derived below would still be valid, with possibly negative distance values for the points having changed side.



Let us consider a point A located on the optical axis of the thick lens at a distance a from its right boundary. A ray passing through A is refracted at the boundary, and the secondary ray intersects the optical axis in a point B located a distance b from the boundary (here B is on the right of the boundary, recall that we take $b > 0$, if B were on the left, we would use $-b$ for the distance). The secondary ray is then refracted at the left boundary of the lens, and the ternary ray finally intersects the optical axis in a point C located at (positive) distance c from the left lens boundary.
Applying the paraxial refraction Eq. (1.5) yields

$$\begin{cases} \dfrac{1}{a} - \dfrac{n}{b} = \dfrac{n-1}{R}, \\ \dfrac{n}{t+b} + \dfrac{1}{c} = \dfrac{n-1}{R}. \end{cases}$$

To establish the thick lens equation, let us first postulate the existence of the focal and principal points of the lens, and compute their positions. Consider the diagram below. This time A is the right focal point F of the lens, and any ray passing through this point emerges from the lens parallel to its optical axis.



This corresponds to $1/c = 0$ or

$$b = \frac{nR}{n-1} - t = \frac{nR - (n-1)t}{n-1}.$$

We can now write

$$\frac{1}{a} = \frac{n}{b} + \frac{n-1}{R} = \frac{n-1}{R}[1 + \frac{nR}{nR-(n-1)t}],$$

or

$$a = \frac{R}{n-1}[1 - \frac{nR}{2nR-(n-1)t}].$$

Now let us assume that the right principal point H of the lens is located on the left of its right boundary, at a (positive) distance d from it. If h is the distance from the optical axis to the point where the first ray enters the lens, and if h' is the distance between the optical axis and the emerging ray, using similar triangles shows that

$$\begin{cases} \dfrac{h'}{d+a} = \dfrac{h}{a} \\ \dfrac{h'}{t+b} = \dfrac{h}{b} \end{cases} \Longrightarrow \frac{d+a}{t+b} = \frac{a}{b} \Longrightarrow d = t\frac{a}{b}.$$

Substituting the values of a and b obtained earlier in this equation shows that

$$d = \frac{Rt}{2Rn - (n-1)t}.$$

The focal length is the distance between H and F and it is thus given by

$$f = d + a = \frac{nR^2}{(n-1)[2Rn-(n-1)t]},$$

or

$$\frac{1}{f} = 2\frac{n-1}{R} - \frac{(n-1)^2}{n}\frac{t}{R^2}.$$

For these values of a, b, c, d, and f, it is now clear that any ray passing through F emerges parallel to the optical axis, and that the emerging ray can be constructed by pretending that the primary ray goes undeflected until it intersects the principal plane passing through H and perpendicular to the optical axis, where refraction turns it into a secondary ray parallel to the axis.

An identical argument allows the construction of the left focal and principal planes. By symmetry, these are located at distances a and d from the left boundary, and the left focal length is the same as the right one.
We gave in Ex. 1.6 a geometric construction of the image P' of a point P given the two focal points F and F' of a thin lens. The same procedure can be used for a thick lens, except for the fact that the ray going through the points P and F (resp. P' and F') is "refracted" into a ray parallel to the optical axis when it crosses the right (resp. left) principal plane instead of the right (resp. left) boundary of the lens (Figure 1.11). It follows immediately that the thin lens equation holds for thick lenses as well, i.e.,

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f},$$

where the origin used to measure z is in the right principal plane instead of at the optical center, and the origin used to measure z' is in the left principal plane.

C H A P T E R 2

Geometric Camera Models

PROBLEMS

2.1. Write formulas for the matrices ${}^A_B\mathcal{R}$ when (B) is deduced from (A) via a rotation of angle θ about the axes $\boldsymbol{i}_{\boldsymbol{A}}$, $\boldsymbol{j}_A$, and $\boldsymbol{k}_A$ respectively.

Solution The expressions for the rotations are obtained by writing the coordinates of the vectors $\boldsymbol{i}_B$, $\boldsymbol{j}_B$ and $\boldsymbol{k}_B$ in the coordinate frame (A). When B is deduced from A by a rotation of angle θ about the axis $\boldsymbol{k}_A$, we obtain

$$ {}^A_B\mathcal{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. $$

Note that ${}^A_B\mathcal{R}$ is of course the inverse of the matrix ${}^B_A\mathcal{R}$ given by Eq. (2.4).
When B is deduced from A by a rotation of angle θ about the axis $\boldsymbol{i}_A$, we have

$$ {}^A_B\mathcal{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}. $$

Finally, when B is deduced from A by a rotation of angle θ about the axis $\boldsymbol{j}_A$, we have

$$ {}^A_B\mathcal{R} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}. $$

2.2. Show that rotation matrices are characterized by the following properties: (a) the inverse of a rotation matrix is its transpose and (b) its determinant is 1.

Solution Let us first show that the two properties are necessary. Consider the rotation matrix

$$ {}^B_A\mathcal{R} = ({}^B\boldsymbol{i}_A \; {}^B\boldsymbol{j}_A \; {}^B\boldsymbol{k}_A). $$

Clearly,

$$ \mathcal{R}^T\mathcal{R} = \begin{pmatrix} {}^B\boldsymbol{i}_A \cdot {}^B\boldsymbol{i}_A & {}^B\boldsymbol{i}_A \cdot {}^B\boldsymbol{j}_A & {}^B\boldsymbol{i}_A \cdot {}^B\boldsymbol{k}_A \\ {}^B\boldsymbol{j}_A \cdot {}^B\boldsymbol{i}_A & {}^B\boldsymbol{j}_A \cdot {}^B\boldsymbol{j}_A & {}^B\boldsymbol{j}_A \cdot {}^B\boldsymbol{k}_A \\ {}^B\boldsymbol{k}_A \cdot {}^B\boldsymbol{i}_A & {}^B\boldsymbol{k}_A \cdot {}^B\boldsymbol{j}_A & {}^B\boldsymbol{k}_A \cdot {}^B\boldsymbol{k}_A \end{pmatrix} = \mathrm{Id}, $$

thus $\mathcal{R}^T$ is the inverse of $\mathcal{R}$. Now $\mathrm{Det}(\mathcal{R}) = ({}^B\boldsymbol{i}_A \times {}^B\boldsymbol{j}_A) \cdot {}^B\boldsymbol{k}_A = 1$ since the vectors $\boldsymbol{i}_A$, $\boldsymbol{j}_A$, and $\boldsymbol{k}_A$ form a right-handed orthonormal basis.
Conversely, suppose that the matrix $\mathcal{R} = \begin{pmatrix} \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} \end{pmatrix}$ verifies the properties $\mathcal{R}^T\mathcal{R} = \mathrm{Id}$ and $\mathrm{Det}(\mathcal{R}) = 1$. Then, reversing the argument used in the "necessary" part of the proof shows that the vectors $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ form a right-handed orthonormal basis of $\mathbb{R}^3$. If we write $\boldsymbol{u} = {}^B\boldsymbol{i}_A$, $\boldsymbol{v} = {}^B\boldsymbol{j}_A$, and $\boldsymbol{w} = {}^B\boldsymbol{k}_A$ for some right-handed orthonormal basis $(B) = (\boldsymbol{i}_B, \boldsymbol{j}_B, \boldsymbol{k}_B)$ of $\mathbb{E}^3$, it is clear that the dot product of any two of the vectors $\boldsymbol{i}_A$, $\boldsymbol{j}_A$, and $\boldsymbol{k}_A$ is the same as the dot product of the corresponding coordinate vectors, e.g., $\boldsymbol{i}_A \cdot \boldsymbol{i}_A = \boldsymbol{u} \cdot \boldsymbol{u} = 1$, and $\boldsymbol{j}_A \cdot \boldsymbol{k}_A = \boldsymbol{v} \cdot \boldsymbol{w} = 0$. By

the same token, $(\boldsymbol{i}_A \times \boldsymbol{j}_A) \cdot \boldsymbol{k}_A = (\boldsymbol{u} \times \boldsymbol{v}) \cdot \boldsymbol{w} = 1$. It follows that the vectors $\boldsymbol{i}_A$, $\boldsymbol{j}_A$, and $\boldsymbol{k}_A$ form a right-handed orthonormal basis of $\mathbb{E}^3$, and that $\mathcal{R}$ is a rotation matrix.

2.3. Show that the set of matrices associated with rigid transformations and equipped with the matrix product forms a group.

Solution Since the set of invertible 4×4 matrices equipped with the matrix product already forms a group, it is sufficient to show that (a) the product of two rigid transformations is also a rigid transformation, and (b) any rigid transformation admits an inverse and this inverse is also a rigid transformation. Let us first prove (a) by considering two rigid transformation matrices

$$\mathcal{T} = \begin{pmatrix} \mathcal{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \quad \mathcal{T}' = \begin{pmatrix} \mathcal{R}' & \boldsymbol{t}' \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

and their product

$$\mathcal{T}'' = \begin{pmatrix} \mathcal{R}'' & \boldsymbol{t}'' \\ \boldsymbol{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathcal{R}\mathcal{R}' & \mathcal{R}\boldsymbol{t}' + \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix}.$$

The matrix $\mathcal{R}''$ is a rotation matrix since $\mathcal{R}''^T\mathcal{R}'' = \mathcal{R}'^T\mathcal{R}^T\mathcal{R}\mathcal{R}' = \mathcal{R}'^T\mathcal{R}' = \mathrm{Id}$ and $\mathrm{Det}(\mathcal{R}'') = \mathrm{Det}(\mathcal{R})\mathrm{Det}(\mathcal{R}') = 1$. Thus $\mathcal{T}''$ is a rigid transformation matrix. To prove (b), note that $\mathcal{T}'' = \mathrm{Id}$ when $\mathcal{R}' = \mathcal{R}^T$ and $\boldsymbol{t}' = -\mathcal{R}^T\boldsymbol{t}$. This shows that any rigid transformation admits an inverse and that this inverse is given by these two equations.

2.4. Let ${}^A\mathcal{T}$ denote the matrix associated with a rigid transformation $\mathcal{T}$ in the coordinate system (A), with

$${}^A\mathcal{T} = \begin{pmatrix} {}^A\mathcal{R} & {}^A\boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix}.$$

Construct the matrix ${}^B\mathcal{T}$ associated with $\mathcal{T}$ in the coordinate system (B) as a function of ${}^A\mathcal{T}$ and the rigid transformation separating (A) and (B).

Solution Let $P' = \mathcal{T}P$ be the image of the point P under the mapping $\mathcal{T}$. Rewriting this equation in the frame (A) yields

$${}^AP' = {}^A\mathcal{T}\,{}^AP,$$

or

$${}^A_B\mathcal{T}\,{}^BP' = {}^A\mathcal{T}\,{}^A_B\mathcal{T}\,{}^BP.$$

In turn, this can be rewritten as

$${}^BP' = {}^A_B\mathcal{T}^{-1}\,{}^A\mathcal{T}\,{}^A_B\mathcal{T}\,{}^BP = {}^B\mathcal{T}\,{}^BP,$$

or, since ${}^A_B\mathcal{T}^{-1} = {}^B_A\mathcal{T}$

$${}^B\mathcal{T} = {}^B_A\mathcal{T}\,{}^A\mathcal{T}\,{}^A_B\mathcal{T}.$$

It follows that an explicit expression for ${}^B\mathcal{T}$ is

$${}^B\mathcal{T} = \begin{pmatrix} {}^B_A\mathcal{R}\,{}^A\mathcal{R}\,{}^A_B\mathcal{R} & {}^B_A\mathcal{R}\,{}^A\mathcal{R}\,{}^AO_B + {}^B_A\mathcal{R}\,{}^A\boldsymbol{t} + {}^BO_A \\ \boldsymbol{0}^T & 1 \end{pmatrix}.$$

2.5. Show that if the coordinate system (B) is obtained by applying to the coordinate system (A) the rigid transformation $\mathcal{T}$, then ${}^{B}P = {}^{A}\mathcal{T}^{-1}\,{}^{A}P$, where ${}^{A}\mathcal{T}$ denotes the matrix representing $\mathcal{T}$ in the coordinate frame (A).

Solution We write

$$\begin{aligned} {}^{A}_{B}\mathcal{T} &= \begin{pmatrix} {}^{A}\boldsymbol{i}_B & {}^{A}\boldsymbol{j}_B & {}^{A}\boldsymbol{k}_B & {}^{A}O_B \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^{A}\mathcal{T}\begin{pmatrix} {}^{A}\boldsymbol{i}_A & {}^{A}\boldsymbol{j}_A & {}^{A}\boldsymbol{k}_A & {}^{A}O_A \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= {}^{A}\mathcal{T}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^{A}\mathcal{T}, \end{aligned}$$

which proves the desired result.

2.6. Show that the rotation of angle θ about the $\boldsymbol{k}$ axis of the frame (F) can be represented by

$$ {}^{F}P' = \mathcal{R}\,{}^{F}P, \quad \text{where} \quad \mathcal{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. $$

Solution Let us write ${}^{F}P = (x, y, z)^T$ and ${}^{F}P' = (x', y', z')^T$. Obviously we must have $z = z'$, the angle between the two vectors (x, y) and (x', y') must be equal to θ, and the norms of these two vectors must be equal. Note that the vector (x, y) is mapped onto the vector $(-y, x)$ by a 90° counterclockwise rotation. Thus we have

$$\begin{cases} \cos\theta = \dfrac{xx' + yy'}{\sqrt{x^2+y^2}\sqrt{x'^2+y'^2}} = \dfrac{xx' + yy'}{x^2+y^2}, \\[2ex] \sin\theta = \dfrac{-yx' + xy'}{\sqrt{x^2+y^2}\sqrt{x'^2+y'^2}} = \dfrac{-yx' + xy'}{x^2+y^2}. \end{cases}$$

Solving this system of linear equations in x' and y' immediately yieds $x' = x\cos\theta - y\sin\theta$ and $y' = x\sin\theta + y\cos\theta$, which proves that we have indeed ${}^{F}P' = \mathcal{R}\,{}^{F}P$.

2.7. Show that the change of coordinates associated with a rigid transformation preserves distances and angles.

Solution Let us consider a fixed coordinate system and identify points of $\mathbb{E}^3$ with their coordinate vectors. Let us also consider three points $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ and their images $\boldsymbol{A}'$, $\boldsymbol{B}'$, and $\boldsymbol{C}'$ under the rigid transformation defined by the rotation matrix $\mathcal{R}$ and the translation vector $\boldsymbol{t}$. The squared distance between $\boldsymbol{A}'$ and $\boldsymbol{B}'$ is

$$|\boldsymbol{B}' - \boldsymbol{A}'|^2 = |\mathcal{R}(\boldsymbol{B} - \boldsymbol{A})|^2 = (\boldsymbol{B} - \boldsymbol{A})^T\mathcal{R}^T\mathcal{R}(\boldsymbol{B} - \boldsymbol{A}) = |\boldsymbol{B} - \boldsymbol{A}|^2,$$

and it follows that rigid transformations preserve distances. Likewise, if θ' denotes the angle between the vectors joining the point $\boldsymbol{A}'$ to the points $\boldsymbol{B}'$ and $\boldsymbol{C}'$, we have

$$\begin{aligned} \cos\theta' &= \frac{(\boldsymbol{B}' - \boldsymbol{A}')\cdot(\boldsymbol{C}' - \boldsymbol{A}')}{|\boldsymbol{B}' - \boldsymbol{A}'|\,|\boldsymbol{C}' - \boldsymbol{A}'|} = \frac{[\mathcal{R}(\boldsymbol{B} - \boldsymbol{A})]\cdot[\mathcal{R}(\boldsymbol{C} - \boldsymbol{A})]}{|\boldsymbol{B} - \boldsymbol{A}|\,|\boldsymbol{C} - \boldsymbol{A}|} \\ &= \frac{(\boldsymbol{B} - \boldsymbol{A})^T R^T\mathcal{R}(\boldsymbol{C} - \boldsymbol{A})}{|\boldsymbol{B} - \boldsymbol{A}|\,|\boldsymbol{C} - \boldsymbol{A}|} = \frac{(\boldsymbol{B} - \boldsymbol{A})\cdot(\boldsymbol{C} - \boldsymbol{A})}{|\boldsymbol{B} - \boldsymbol{A}|\,|\boldsymbol{C} - \boldsymbol{A}|} = \cos\theta, \end{aligned}$$

where θ is the angle between the vectors joining the point $\boldsymbol{A}$ to the points $\boldsymbol{B}$ and $\boldsymbol{C}$. It follows that rigid transformations also preserve angles (to be rigorous, we should also show that the sine of θ is also preserved). Note that the translation part of the rigid transformation is irrelevent in both cases.

2.8. Show that when the camera coordinate system is skewed and the angle θ between the two image axes is not equal to 90 degrees, then Eq. (2.11) transforms into Eq. (2.12).

Solution Let us denote by $(\hat{u}, \hat{v})$ the normalized coordinate system for the image plane centered in the projection C_0 of the optical center (see Figure 2.8). Let us also denote by $(\tilde{u}, \tilde{v})$ a skew coordinate centered in C_0 with unit basis vectors and a skew angle equal to θ. Overlaying the orthogonal and skew coordinate systems immediately reveals that $\hat{v}/\tilde{v} = \sin\theta$ and $\hat{u} = \tilde{u} + \hat{v}\cot\theta$, or

$$\begin{cases} \tilde{u} = \hat{u} - \hat{v}\cot\theta = \dfrac{x}{z} - \cot\theta\dfrac{y}{z} \\ \tilde{v} = \dfrac{1}{\sin\theta}\hat{v} = \dfrac{1}{\sin\theta}\dfrac{y}{z}. \end{cases}$$

Taking into account the actual position of the image center and the camera magnifications yields

$$\begin{cases} u = \alpha\tilde{u} + u_0 = \alpha\dfrac{x}{z} - \alpha\cot\theta\dfrac{y}{z} + u_0, \\ v = \beta\tilde{v} + v_0 = \dfrac{\beta}{\sin\theta}\dfrac{y}{z} + v_0, \end{cases}$$

which is the desired result.

2.9. Let $\boldsymbol{O}$ denote the *homogeneous* coordinate vector of the optical center of a camera in some reference frame, and let $\mathcal{M}$ denote the corresponding perspective projection matrix. Show that $\mathcal{M}\boldsymbol{O} = \mathbf{0}$.

Solution As shown by Eq. (2.15), the most general form of the perspective projection matrix in some world coordinate system (W) is $\mathcal{M} = \mathcal{K}\begin{pmatrix} {}^C_W\mathcal{R} & {}^CO_W \end{pmatrix}$. Here, ${}^CO_W$ is the *non-homogeneous* coordinate vector of the origin of (W) in the normalized coordinate system (C) attached to the camera. On the other hand, $\boldsymbol{O}$ is by definition the *homogeneous* coordinate vector of the origin of (C)—that is, the camera's optical center—in the world coordiante system, so $\boldsymbol{O}^T = ({}^WO_C^T, 1)$. Thus

$$\mathcal{M}\boldsymbol{O} = \mathcal{K}\begin{pmatrix} {}^C_W\mathcal{R} & {}^CO_W \end{pmatrix}\begin{pmatrix} {}^WO_C \\ 1 \end{pmatrix} = \mathcal{K}({}^C_W\mathcal{R}\,{}^WO_C + {}^CO_W) = \mathcal{K}{}^CO_C = \mathcal{K}\mathbf{0} = \mathbf{0}.$$

2.10. Show that the conditions of Theorem 1 are necessary.

Solution As noted in the chapter itself, according to Eq. 2.15, we have $\mathcal{A} = \mathcal{K}\mathcal{R}$, thus the determinants of $\mathcal{A}$ and $\mathcal{K}$ are the same and $\mathcal{A}$ is nonsingular. Now, according to Eq. (2.17), we have

$$\begin{cases} \boldsymbol{a}_1 = \alpha\boldsymbol{r}_1 - \alpha\cot\theta\boldsymbol{r}_2 + u_0\boldsymbol{r}_3, \\ \boldsymbol{a}_2 = \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2 + v_0\boldsymbol{r}_3, \\ \boldsymbol{a}_3 = \boldsymbol{r}_3, \end{cases}$$

where $\boldsymbol{r}_1^T$, $\boldsymbol{r}_2^T$ and $\boldsymbol{r}_3^T$ denote the row vectors of the rotation matrix $\mathcal{R}$. It follows that

$$(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = (-\alpha \boldsymbol{r}_2 + \alpha \cot\theta \boldsymbol{r}_1) \cdot (\frac{\beta}{\sin\theta}\boldsymbol{r}_1) = \alpha\beta\cos\theta,$$

thus $(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0$ implies that $\cos\theta = 0$ and the cameras as zero skew. Finally, we have

$$\begin{aligned} |\boldsymbol{a}_1 \times \boldsymbol{a}_3|^2 - |\boldsymbol{a}_2 \times \boldsymbol{a}_3|^2 &= |-\alpha \boldsymbol{r}_2 + \alpha\cot\theta \boldsymbol{r}_1|^2 - |\frac{\beta}{\sin\theta}\boldsymbol{r}_1|^2 \\ &= \alpha^2(1 + \cot^2\theta) - \frac{\beta^2}{\sin^2\theta} = \frac{\alpha^2 - \beta^2}{\sin^2\theta}. \end{aligned}$$

Thus $|\boldsymbol{a}_1 \times \boldsymbol{a}_3|^2 = |\boldsymbol{a}_2 \times \boldsymbol{a}_3|^2$ implies that $\alpha^2 = \beta^2$, i.e., that the camera has unit aspect ratio.

2.11. Show that the conditions of Theorem 1 are sufficient. Note that the statement of this theorem is a bit different from the corresponding theorems in Faugeras (1993) and Heyden (1995), where the condition $\text{Det}(\mathcal{A}) \neq 0$ is replaced by $\boldsymbol{a}_3 \neq 0$. Of course, $\text{Det}(\mathcal{A}) \neq 0$ implies $\boldsymbol{a}_3 \neq 0$.

Solution We follow here the procedure for the recovery of a camera's intrinsic and extrinsic parameters given in Section 3.2.1 of the next chapter. The conditions of Theorem 1 ensure via Eq. (3.13) that this procedure succeeds and yields the correct intrinsic parameters. In particular, when the determinant of $\mathcal{M}$ is nonzero, the vectors $\boldsymbol{a}_i$ are linearly independent, and their pairwise cross-products are nonzero, ensuring that all terms in Eq. (3.13) are well defined. Adding the condition $(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0$ gives $\cos\theta$ a value of zero in the that equation, yielding a zero-skew camera. Finally, adding the condition $|\boldsymbol{a}_1 \times \boldsymbol{a}_3|^2 = |\boldsymbol{a}_2 \times \boldsymbol{a}_3|^2$ gives the two magnifications equal values, corresponding to a unit aspect-ratio.

2.12. If ${}^A\boldsymbol{\Pi}$ denotes the homogeneous coordinate vector of a plane Π in the coordinate frame (A), what is the homogeneous coordinate vector ${}^B\boldsymbol{\Pi}$ of Π in the frame (B)?

Solution Let ${}^A_B\mathcal{T}$ denote the matrix representing the change of coordinates between the frames (B) and (A) such that ${}^AP = {}^A_B\mathcal{T}\,{}^BP$. We have, for any point P in the plane Π,

$$0 = {}^A\boldsymbol{\Pi}^T\,{}^AP = {}^A\boldsymbol{\Pi}^T\,{}^A_B\mathcal{T}\,{}^BP = [{}^A_B\mathcal{T}^T\,{}^A\boldsymbol{\Pi}]^T\,{}^BP = {}^B\boldsymbol{\Pi}^T\,{}^BP.$$

Thus ${}^B\boldsymbol{\Pi} = {}^A_B\mathcal{T}^T\,{}^A\boldsymbol{\Pi}$.

2.13. If ${}^A\mathcal{Q}$ denotes the symmetric matrix associated with a quadric surface in the coordinate frame (A), what is the symmetric matrix ${}^B\mathcal{Q}$ associated with this surface in the frame (B)?

Solution Let ${}^A_B\mathcal{T}$ denote the matrix representing the change of coordinates between the frames (B) and (A) such that ${}^AP = {}^A_B\mathcal{T}\,{}^BP$. We have, for any point P on the quadric surface,

$$0 = {}^AP^T\,{}^A\mathcal{Q}\,{}^AP = ({}^BP^T\,{}^A_B\mathcal{T}^T)\,{}^A\mathcal{Q}\,({}^A_B\mathcal{T}\,{}^BP) = {}^BP^T\,{}^B\mathcal{Q}\,{}^BP.$$

Thus ${}^B\mathcal{Q} = {}^A_B\mathcal{T}^T\,{}^A\mathcal{Q}\,{}^A_B\mathcal{T}$. This matrix is symmetric by construction.

2.14. Prove Theorem 2.

Solution Let us consider an affine projection matrix $\mathcal{M} = \begin{pmatrix}\mathcal{A} & \boldsymbol{b}\end{pmatrix}$, and let us first show that it can be written as a general weak-perspective projection matrix as defined by Eq. (2.20), i.e.,

$$\mathcal{M} = \frac{1}{z_r}\begin{pmatrix} k & s \\ 0 & 1 \end{pmatrix}\begin{pmatrix}\mathcal{R}_2 & \boldsymbol{t}_2\end{pmatrix}.$$

Writing the entries of the two matrices explicitly yields

$$\begin{pmatrix} \boldsymbol{a}_1^T & b_1 \\ \boldsymbol{a}_2^T & b_2 \end{pmatrix} = \frac{1}{z_r}\begin{pmatrix} k\boldsymbol{r}_1^T + s\boldsymbol{r}_2^T & kt_x + st_y \\ \boldsymbol{r}_2^T & t_y \end{pmatrix}.$$

Assuming that z_r is positive, it follows that $z_r = 1/|\boldsymbol{a}_2|$, $\boldsymbol{r}_2 = z_r\boldsymbol{a}_2$, $t_y = z_r b_2$. In addition, $s = z_r^2(\boldsymbol{a}_1 \cdot \boldsymbol{a}_2)$, so $k\boldsymbol{r}_1 = z_r(\boldsymbol{a}_1 - s\boldsymbol{a}_2)$. Assuming that k is positive, we obtain $k = z_r|\boldsymbol{a}_1 - s\boldsymbol{a}_2|$, and $\boldsymbol{r}_1 = (z_r/k)(\boldsymbol{a}_1 - s\boldsymbol{a}_2)$. Finally, $t_x = (z_r b_1 - kt_x)/s$. Picking a negative value for z_r and/or k yields a total of four solutions.
Let us now show that $\mathcal{M}$ can be written as a paraperspective projection matrix as defined by Eq. (2.21) with $k = 1$ and $s = 0$, i.e.,

$$\mathcal{M} = \frac{1}{z_r}\left(\begin{pmatrix} 1 & 0 & -x_r/z_r \\ 0 & 1 & -y_r/z_r \end{pmatrix}\mathcal{R} \quad \boldsymbol{t}_2\right).$$

We start by showing that a paraperspective projection matrix can indeed always be written in this form before showing that any affine projection matrix can be written in this form as well.
Recall that x_r, y_r and z_r denote the coordinates of the reference point R in the normalized camera coordinate system. The two elementary projection stages $P \to P' \to p$ can be written in this frame as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x' \\ y' \\ z_r \end{pmatrix} \longrightarrow \begin{pmatrix} \hat{u} \\ \hat{v} \\ 1 \end{pmatrix} = \begin{pmatrix} x'/z_r \\ y'/z_r \\ 1 \end{pmatrix},$$

where x' and y' are the coordinates of the point P'. Writing that $\overrightarrow{PP'}$ is parallel to $\overrightarrow{OQ}$ or equivalently that $\overrightarrow{PP'} \times \overrightarrow{OQ} = \boldsymbol{0}$ yields

$$\begin{cases} x' = \dfrac{1}{z_r}(x - \dfrac{x_r}{z_r}z + x_r), \\ y' = \dfrac{1}{z_r}(y - \dfrac{y_r}{z_r}z + y_r). \end{cases}$$

The normalized image coordinates of the point p can thus be written as

$$\begin{pmatrix} \hat{u} \\ \hat{v} \\ 1 \end{pmatrix} = \frac{1}{z_r}\begin{pmatrix} 1 & 0 & -x_r/z_r & x_r \\ 0 & 1 & -y_r/z_r & y_r \\ 0 & 0 & 0 & z_r \end{pmatrix}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Introducing once again the intrinsic and extrinsic parameters $\mathcal{K}_2$, $\boldsymbol{p}_0$, $\mathcal{R}$ and $\boldsymbol{t}$ of the camera and using the fact that z_r is independent of the point P gives the general form of the projection equations, i.e.,

$$\boldsymbol{p} = \frac{1}{z_r}\begin{pmatrix}\mathcal{K}_2 & \boldsymbol{p}_0\end{pmatrix}\begin{pmatrix} 1 & 0 & -x_r/z_r & x_r \\ 0 & 1 & -y_r/z_r & y_r \\ 0 & 0 & 0 & z_r \end{pmatrix}\begin{pmatrix} \mathcal{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix}\begin{pmatrix} \boldsymbol{P} \\ 1 \end{pmatrix},$$

which can indeed be written as an instance of the general affine projection equation (2.19) with

$$\begin{cases} \mathcal{A} = \dfrac{1}{z_r}\mathcal{K}_2 \begin{pmatrix} 1 & 0 & -x_r/z_r \\ 0 & 1 & -y_r/z_r \end{pmatrix} \mathcal{R}, \\ \boldsymbol{b} = \dfrac{1}{z_r}\mathcal{K}_2 \left[\boldsymbol{t}_2 + (1 - \dfrac{t_z}{z_r}) \begin{pmatrix} x_r \\ y_r \end{pmatrix} \right] + \boldsymbol{p}_0. \end{cases}$$

The translation parameters are coupled to the camera intrinsic parameters and the position of the reference point in the expression of $\boldsymbol{b}$. As in the weak-perspective case, we are free to change the position of the camera relative to the origin of the world coordinate system to simplify this expression. In particular, we can choose $t_z = z_r$ and reset the value of $\boldsymbol{t}_2$ to $\boldsymbol{t}_2 - z_r \mathcal{K}_2^{-1} \boldsymbol{p}_0$, so the value of $\boldsymbol{b}$ becomes $\frac{1}{z_r}\mathcal{K}_2 \boldsymbol{t}_2$. Now, observing that the value of $\mathcal{A}$ does not change when $\mathcal{K}_2$ and (x_r, y_r, z_r) are respectively replaced by $\lambda\mathcal{K}_2$ and $\lambda(x_r, y_r, z_r)$ allows us to rewrite the projection matrix $\mathcal{M} = \begin{pmatrix} \mathcal{A} & \boldsymbol{b} \end{pmatrix}$ as

$$\mathcal{M} = \frac{1}{z_r} \begin{pmatrix} k & s \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 & -x_r/z_r \\ 0 & 1 & -y_r/z_r \end{pmatrix} \mathcal{R} \quad \boldsymbol{t}_2 \right).$$

Let us now show that any affine projection matrix can be written in this form. We can rewrite explicitly the entries of the two matrices of interest as:

$$\begin{pmatrix} \boldsymbol{a}_1^T & b_1 \\ \boldsymbol{a}_2^T & b_2 \end{pmatrix} = \frac{1}{z_r} \begin{pmatrix} \boldsymbol{r}_1^T - \dfrac{x_r}{z_r}\boldsymbol{r}_3^T & t_x \\ \boldsymbol{r}_2^T - \dfrac{y_r}{z_r}\boldsymbol{r}_3^T & t_y \end{pmatrix}.$$

Since $\boldsymbol{r}_1^T$, $\boldsymbol{r}_2^T$, and $\boldsymbol{r}_3^T$ are the rows of a rotation matrix, these vectors are orthogonal and have unit norm, and it follows that

$$\begin{cases} 1 + \lambda^2 = z_r^2 |\boldsymbol{a}_1|^2, \\ 1 + \mu^2 = z_r^2 |\boldsymbol{a}_2|^2, \\ \lambda\mu = z_r^2 (\boldsymbol{a}_1 \cdot \boldsymbol{a}_2), \end{cases}$$

where $\lambda = x_r/z_r$ and $\mu = y_r/z_r$. Eliminating z_r among these equations yields

$$\begin{cases} \dfrac{\lambda\mu}{1+\lambda^2} = c_1, \\ \dfrac{\lambda\mu}{1+\mu^2} = c_2, \end{cases}$$

where $c_1 = (\boldsymbol{a}_1 \cdot \boldsymbol{a}_2)/|\boldsymbol{a}_1|^2$ and $c_2 = (\boldsymbol{a}_1 \cdot \boldsymbol{a}_2)/|\boldsymbol{a}_2|^2$. It follows from the first equation that $\mu = c_1(1 + \lambda^2)/\lambda$, and substituting in the second equation yields, after some simple algebraic manipulation,

$$(1 - c_1 c_2)\lambda^4 + [1 - c_1 c_2 - \frac{c_2}{c_1}(1 + c_1^2)]\lambda^2 - c_1 c_2 = 0.$$

This is a quadratic equation in λ^2. Note that $c_1 c_2$ is the squared cosine of the angle between $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, so the constant and quadratic term have opposite signs. In particular, our quadratic equation always admits two real roots with opposite

signs. The positive root is the only root of interest, and it yields two opposite values for λ. Once λ is known, μ is of course determined uniquely, and so is z_r^2. It follows that there are four possible solutions for the triple (x_r, y_r, z_r). For each of these solutions, we have $t_x = z_r b_1$ and $t_y = z_r b_2$.
We finally determine $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$ by defining $\boldsymbol{a}_3 = \boldsymbol{a}_1 \times \boldsymbol{a}_2$ and noting that $\lambda \boldsymbol{r}_1 + \mu \boldsymbol{r}_2 + \boldsymbol{r}_3 = z_r^2 \boldsymbol{a}_3$. In particular, we can write

$$z_r \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & z_r \boldsymbol{a}_3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & \lambda \\ 0 & 1 & \mu \\ -\lambda & -\mu & 1 \end{pmatrix}.$$

Multiplying both sides of this equation on the right by the inverse of the rightmost matrix yields

$$\begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{pmatrix} = \frac{z_r}{1+\lambda^2+\mu^2} \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & z_r \boldsymbol{a}_3 \end{pmatrix} \begin{pmatrix} 1+\mu^2 & -\lambda\mu & -\lambda \\ -\lambda\mu & 1+\lambda^2 & -\mu \\ \lambda & \mu & 1 \end{pmatrix},$$

or

$$\begin{cases} \boldsymbol{r}_1 = \dfrac{z_r}{1+\lambda^2+\mu^2}[(1+\mu^2)\boldsymbol{a}_1 - \lambda\mu \boldsymbol{a}_2 + \lambda z_r \boldsymbol{a}_3], \\ \boldsymbol{r}_2 = \dfrac{z_r}{1+\lambda^2+\mu^2}[-\lambda\mu \boldsymbol{a}_1 + (1+\lambda^2)\boldsymbol{a}_2 + \mu z_r \boldsymbol{a}_3], \\ \boldsymbol{r}_3 = \dfrac{z_r}{1+\lambda^2+\mu^2}[-\lambda \boldsymbol{a}_1 - \mu \boldsymbol{a}_2 + z_r \boldsymbol{a}_3]. \end{cases}$$

2.15. Line Plücker coordinates. The *exterior product* of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{R}^4$ is defined by

$$\boldsymbol{u} \wedge \boldsymbol{v} \stackrel{\text{def}}{=} \begin{pmatrix} u_1 v_2 - u_2 v_1 \\ u_1 v_3 - u_3 v_1 \\ u_1 v_4 - u_4 v_1 \\ u_2 v_3 - u_3 v_2 \\ u_2 v_4 - u_4 v_2 \\ u_3 v_4 - u_4 v_3 \end{pmatrix}.$$

Given a fixed coordinate system and the (homogeneous) coordinates vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ associated with two points A and B in $\mathbb{E}^3$, the vector $\boldsymbol{L} = \boldsymbol{A} \wedge \boldsymbol{B}$ is called the vector of Plücker coordinates of the line joining A to B.

(a) Let us write $\boldsymbol{L} = (L_1, L_2, L_3, L_4, L_5, L_6)^T$ and denote by O the origin of the coordinate system and by H its projection onto L. Let us also identify the vectors $\overrightarrow{OA}$ and $\overrightarrow{OB}$ with their non-homogeneous coordinate vectors. Show that $\overrightarrow{AB} = -(L_3, L_5, L_6)^T$ and $\overrightarrow{OA} \times \overrightarrow{OB} = \overrightarrow{OH} \times \overrightarrow{AB} = (L_4, -L_2, L_1)^T$. Conclude that the Plücker coordinates of a line obey the quadratic constraint $L_1 L_6 - L_2 L_5 + L_3 L_4 = 0$.

(b) Show that changing the position of the points A and B along the line L only changes the overall scale of the vector $\boldsymbol{L}$. Conclude that Plücker coordinates are homogeneous coordinates.

(c) Prove that the following identity holds of any vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, and $\boldsymbol{t}$ in $\mathbb{R}^4$:

$$(\boldsymbol{x} \wedge \boldsymbol{y}) \cdot (\boldsymbol{z} \wedge \boldsymbol{t}) = (\boldsymbol{x} \cdot \boldsymbol{z})(\boldsymbol{y} \cdot \boldsymbol{t}) - (\boldsymbol{x} \cdot \boldsymbol{t})(\boldsymbol{y} \cdot \boldsymbol{z}).$$

(d) Use this identity to show that the mapping between a line with Plücker coordinate vector $\boldsymbol{L}$ and its image l with homogeneous coordinates $\boldsymbol{l}$ can be represented by

$$\rho \boldsymbol{l} = \tilde{\mathcal{M}} \boldsymbol{L}, \quad \text{where} \quad \tilde{\mathcal{M}} \stackrel{\text{def}}{=} \begin{pmatrix} (\boldsymbol{m}_2 \wedge \boldsymbol{m}_3)^T \\ (\boldsymbol{m}_3 \wedge \boldsymbol{m}_1)^T \\ (\boldsymbol{m}_1 \wedge \boldsymbol{m}_2)^T \end{pmatrix}, \tag{2.1}$$

and $\boldsymbol{m}_1^T$, $\boldsymbol{m}_2^T$, and $\boldsymbol{m}_3^T$ denote as before the rows of $\mathcal{M}$ and ρ is an appropriate scale factor.

Hint: Consider a line L joining two points A and B and denote by a and b the projections of these two points, with homogeneous coordinates $\boldsymbol{a}$ and $\boldsymbol{b}$. Use the fact that the points a and b lie on l, thus if $\boldsymbol{l}$ denote the homogeneous coordinate vector of this line, we must have $\boldsymbol{l} \cdot \boldsymbol{a} = \boldsymbol{l} \cdot \boldsymbol{b} = 0$.

(e) Given a line L with Plücker coordinate vector $\boldsymbol{L} = (L_1, L_2, L_3, L_4, L_5, L_6)^T$ and a point P with homogeneous coordinate vector $\boldsymbol{P}$, show that a necessary and sufficient condition for P to lie on L is that

$$\mathcal{L}\boldsymbol{P} = 0, \quad \text{where} \quad \mathcal{L} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & L_6 & -L_5 & L_4 \\ -L_6 & 0 & L_3 & -L_2 \\ L_5 & -L_3 & 0 & L_1 \\ -L_4 & L_2 & -L_1 & 0 \end{pmatrix}.$$

(f) Show that a necessary and sufficient condition for the line L to lie in the plane Π with homogeneous coordinate vector $\boldsymbol{\Pi}$ is that

$$\mathcal{L}^* \boldsymbol{\Pi} = 0, \quad \text{where} \quad \mathcal{L}^* \stackrel{\text{def}}{=} \begin{pmatrix} 0 & L_1 & L_2 & L_3 \\ -L_1 & 0 & L_4 & L_5 \\ -L_2 & -L_4 & 0 & L_6 \\ -L_3 & -L_5 & -L_6 & 0 \end{pmatrix}.$$

Solution

(a) If $\boldsymbol{A} = (a_1, a_2, a_3, 1)^T$ and $\boldsymbol{B} = (b_1, b_2, b_3, 1)^T$, we have

$$\boldsymbol{L} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{pmatrix} = \boldsymbol{A} \wedge \boldsymbol{B} = \begin{pmatrix} a_1 b_2 - a_2 b_1 \\ a_1 b_3 - a_3 b_1 \\ a_1 - b_1 \\ a_2 b_3 - a_3 b_2 \\ a_2 - b_2 \\ a_3 - b_3 \end{pmatrix},$$

thus $\overrightarrow{AB} = -(L_3, L_5, L_6)^T$ and $\overrightarrow{OA} \times \overrightarrow{OB} = (L_4, -L_2, L_1)^T$. In addition, we have

$$\overrightarrow{OA} \times \overrightarrow{OB} = (\overrightarrow{OH} + \overrightarrow{HA}) \times (\overrightarrow{OH} + \overrightarrow{HB}) = \overrightarrow{OH} \times \overrightarrow{AB}$$

since $\overrightarrow{HA}$ and $\overrightarrow{HB}$ are parallel.

Since the vectors $\overrightarrow{AB}$ and $\overrightarrow{OA} \times \overrightarrow{OB}$ are orthogonal, it follows immediately that their dot product $L_1 L_6 - L_2 L_5 + L_3 L_4$ is equal to zero.

(b) Replacing A and B by any other two points C and D on the same line only scales the Plücker coordinates of this lines since we can always write $\overrightarrow{CD} = \overrightarrow{AB}$

and, according to (a), this means that $\boldsymbol{C} \wedge \boldsymbol{D} = \lambda \boldsymbol{A} \wedge \boldsymbol{B}$. Thus the Plücker coordinates of a line are uniquely defined up to scale, independently of the choice of the points chosen to represent this line. In other words, they are homogeneous coordinates for this line.

(c) Writing

$$\begin{aligned}(\boldsymbol{x} \wedge \boldsymbol{y}) \cdot (\boldsymbol{z} \wedge \boldsymbol{t}) &= (x_1 y_2 - x_2 y_1)(z_1 t_2 - z_2 t_1) + (x_1 y_3 - x_3 y_1)(z_1 t_3 - z_3 t_1) \\ &+ (x_1 y_4 - x_4 y_1)(z_1 t_4 - z_4 t_1) + (x_2 y_3 - x_3 y_2)(z_2 t_3 - z_3 t_2) \\ &+ (x_2 y_4 - x_4 y_2)(z_2 t_4 - z_4 t_2) + (x_3 y_4 - x_4 y_3)(z_3 t_4 - z_4 t_3) \\ &= (x_1 z_1 + x_2 z_2 + x_3 z_3 + x_4 z_4)(y_1 t_1 + y_2 t_2 + y_3 t_3 + y_4 t_4) \\ &- (x_1 t_1 + x_2 t_2 + x_3 t_3 + x_4 t_4)(y_1 z_1 + y_2 z_2 + y_3 z_3 + y_4 z_4) \\ &= (\boldsymbol{x} \cdot \boldsymbol{z})(\boldsymbol{y} \cdot \boldsymbol{t}) - (\boldsymbol{x} \cdot \boldsymbol{t})(\boldsymbol{y} \cdot \boldsymbol{z})\end{aligned}$$

proves the identity.

(d) Let us consider the line L passing through the points A and B with homogeneous coordinates $\boldsymbol{L}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ and denote by l the line's projection and by a and b the images of A and B, with homogeneous coordinate vectors $\boldsymbol{l}$, $\boldsymbol{a}$ and $\boldsymbol{b}$. The identity proven in (c) allows to write

$$\begin{aligned}\tilde{\mathcal{M}}\boldsymbol{L} &= \begin{pmatrix}(\boldsymbol{m}_2 \wedge \boldsymbol{m}_3)^T \\ (\boldsymbol{m}_3 \wedge \boldsymbol{m}_1)^T \\ (\boldsymbol{m}_1 \wedge \boldsymbol{m}_2)^T\end{pmatrix}(\boldsymbol{A} \wedge \boldsymbol{B}) = \begin{pmatrix}(\boldsymbol{m}_2 \wedge \boldsymbol{m}_3) \cdot (\boldsymbol{A} \wedge \boldsymbol{B}) \\ (\boldsymbol{m}_3 \wedge \boldsymbol{m}_1) \cdot (\boldsymbol{A} \wedge \boldsymbol{B}) \\ (\boldsymbol{m}_1 \wedge \boldsymbol{m}_2) \cdot (\boldsymbol{A} \wedge \boldsymbol{B})\end{pmatrix} \\ &= \begin{pmatrix}(\boldsymbol{m}_2 \cdot \boldsymbol{A})(\boldsymbol{m}_3 \cdot \boldsymbol{B}) - (\boldsymbol{m}_2 \cdot \boldsymbol{B})(\boldsymbol{m}_3 \cdot \boldsymbol{A}) \\ (\boldsymbol{m}_3 \cdot \boldsymbol{A})(\boldsymbol{m}_1 \cdot \boldsymbol{B}) - (\boldsymbol{m}_3 \cdot \boldsymbol{B})(\boldsymbol{m}_1 \cdot \boldsymbol{A}) \\ (\boldsymbol{m}_1 \cdot \boldsymbol{A})(\boldsymbol{m}_2 \cdot \boldsymbol{B}) - (\boldsymbol{m}_1 \cdot \boldsymbol{B})(\boldsymbol{m}_2 \cdot \boldsymbol{A})\end{pmatrix} \\ &= \boldsymbol{a} \times \boldsymbol{b}.\end{aligned}$$

Since $\boldsymbol{l}$ must be orthogonal to $\boldsymbol{a}$ and $\boldsymbol{b}$ we can thus write $\rho \boldsymbol{l} = \tilde{\mathcal{M}}\boldsymbol{L}$.

(e) Let us assume that $\boldsymbol{L} = \boldsymbol{A} \wedge \boldsymbol{B}$. Since $\boldsymbol{P}^T = (\overrightarrow{OP}^T, 1)$ we can write

$$\begin{aligned}\mathcal{L}\boldsymbol{P} &= \begin{pmatrix}0 & -\overrightarrow{AB}_3 & \overrightarrow{AB}_2 & (\overrightarrow{OA} \times \overrightarrow{OB})_1 \\ \overrightarrow{AB}_3 & 0 & -\overrightarrow{AB}_1 & (\overrightarrow{OA} \times \overrightarrow{OB})_2 \\ -\overrightarrow{AB}_2 & \overrightarrow{AB}_1 & 0 & (\overrightarrow{OA} \times \overrightarrow{OB})_3 \\ -(\overrightarrow{OA} \times \overrightarrow{OB})_1 & -(\overrightarrow{OA} \times \overrightarrow{OB})_2 & -(\overrightarrow{OA} \times \overrightarrow{OB})_3 & 0\end{pmatrix}\begin{pmatrix}\overrightarrow{OP} \\ 1\end{pmatrix} \\ &= \begin{pmatrix}(\overrightarrow{AB} \times \overrightarrow{OP})_1 + (\overrightarrow{OA} \times \overrightarrow{OB})_1 \\ (\overrightarrow{AB} \times \overrightarrow{OP})_2 + (\overrightarrow{OA} \times \overrightarrow{OB})_2 \\ (\overrightarrow{AB} \times \overrightarrow{OP})_3 + (\overrightarrow{OA} \times \overrightarrow{OB})_3 \\ -(\overrightarrow{OA} \times \overrightarrow{OB}) \cdot \overrightarrow{OP}\end{pmatrix} = \begin{pmatrix}\overrightarrow{AB} \times \overrightarrow{BP} \\ -(\overrightarrow{OH} \times \overrightarrow{AB}) \cdot \overrightarrow{OP}\end{pmatrix}.\end{aligned}$$

Thus a necessary and sufficient condition for $\mathcal{L}\boldsymbol{P}$ to be equal to the zero vector is that $\overrightarrow{AP}$ be parallel to $\overrightarrow{BP}$ and that the three points A, B, and P be collinear, or that P lie on the line L.

(f) Let us write $\boldsymbol{\Pi}^T = (\boldsymbol{n}^T, -d)$, where $\boldsymbol{n}$ is the unit normal to the plane Π and d is the distance between the origin and the plane. We have

$$\begin{aligned}\mathcal{L}^*\boldsymbol{\Pi} &= \begin{pmatrix}0 & (\overrightarrow{OA} \times \overrightarrow{OB})_3 & -(\overrightarrow{OA} \times \overrightarrow{OB})_2 & -\overrightarrow{AB}_1 \\ -(\overrightarrow{OA} \times \overrightarrow{OB})_3 & 0 & (\overrightarrow{OA} \times \overrightarrow{OB})_1 & -\overrightarrow{AB}_2 \\ (\overrightarrow{OA} \times \overrightarrow{OB})_2 & -(\overrightarrow{OA} \times \overrightarrow{OB})_1 & 0 & -\overrightarrow{AB}_3 \\ \overrightarrow{AB}_1 & \overrightarrow{AB}_2 & \overrightarrow{AB}_3 & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{n} \\ -d\end{pmatrix} \\ &= \begin{pmatrix}\boldsymbol{n} \times (\overrightarrow{OA} \times \overrightarrow{OB}) + d\overrightarrow{AB} \\ \boldsymbol{n} \cdot \overrightarrow{AB}\end{pmatrix} = \begin{pmatrix}\boldsymbol{n} \times (\overrightarrow{OH} \times \overrightarrow{AB}) + d\overrightarrow{AB} \\ \boldsymbol{n} \cdot \overrightarrow{AB}\end{pmatrix}.\end{aligned}$$

Now, it is easy to show that $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} - (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{c}$ for any vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ of $\mathbb{R}^3$. Hence,

$$\mathcal{L}^*\mathbf{\Pi} = \begin{pmatrix} (\boldsymbol{n} \cdot \overrightarrow{AB})\overrightarrow{OH} + (d - \boldsymbol{n} \cdot \overrightarrow{OH})\overrightarrow{AB} \\ \boldsymbol{n} \cdot \overrightarrow{AB} \end{pmatrix},$$

and a necessary and sufficient condition for $\mathcal{L}^*\mathbf{\Pi}$ to be the zero vector is that the vector $\overrightarrow{AB}$ lie in a plane parallel to Π (condition $\boldsymbol{n} \cdot \overrightarrow{AB} = 0$), located at a distance d from the origin (condition $(\boldsymbol{n} \cdot \overrightarrow{AB})\overrightarrow{OH} + (d - \boldsymbol{n} \cdot \overrightarrow{OH})\overrightarrow{AB} = 0 \Longrightarrow d = \boldsymbol{n} \cdot \overrightarrow{OH}$), or equivalently, that L lie in the plane Π.

C H A P T E R 3

Geometric Camera Calibration

PROBLEMS

3.1. Show that the vector $\boldsymbol{x}$ that minimizes $|\mathcal{U}\boldsymbol{x}|^2$ under the constraint $|\mathcal{V}\boldsymbol{x}|^2 = 1$ is the (appropriately scaled) generalized eigenvector associated with the minimum generalized eigenvalue of the symmetric matrices $\mathcal{U}^T\mathcal{U}$ and $\mathcal{V}^T\mathcal{V}$.
Hint: First show that the minimum sought is reached at $\boldsymbol{x} = \boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is the (unconstrained) minimum of the error $E(\boldsymbol{x}) = |\mathcal{U}\boldsymbol{x}|^2/|\mathcal{V}\boldsymbol{x}|^2$ such that $|\mathcal{V}\boldsymbol{x}_0|^2 = 1$. (Note that since $E(\boldsymbol{x})$ is obviously invariant under scale changes, so are its extrema, and we are free to fix the norm of $|\mathcal{V}\boldsymbol{x}_0|^2$ arbitrarily. Note also that the minimum must be taken over all values of $\boldsymbol{x}$ such that $\mathcal{V}\boldsymbol{x} \neq 0$.)

Solution By definition of $\boldsymbol{x}_0$, we have

$$\forall \boldsymbol{x}, \quad \mathcal{V}\boldsymbol{x} \neq \mathbf{0} \Longrightarrow \frac{|\mathcal{U}\boldsymbol{x}|^2}{|\mathcal{V}\boldsymbol{x}|^2} \geq \frac{|\mathcal{U}\boldsymbol{x}_0|^2}{|\mathcal{V}\boldsymbol{x}_0|^2} = |\mathcal{U}\boldsymbol{x}_0|^2.$$

In particular,

$$\forall \boldsymbol{x}, |\mathcal{V}\boldsymbol{x}|^2 = 1 \Longrightarrow |\mathcal{U}\boldsymbol{x}|^2 \geq |\mathcal{U}\boldsymbol{x}_0|^2.$$

Since, by definition, $|\mathcal{V}\boldsymbol{x}_0|^2 = 1$, it follows that $\boldsymbol{x}_0$ is indeed the minimum of the constrained minimization problem.
Let us now compute $\boldsymbol{x}_0$. The gradient of $E(\boldsymbol{x})$ must vanish at its extrema. Its value is

$$\nabla E = \frac{1}{|\mathcal{V}\boldsymbol{x}|^2}\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^T\mathcal{U}^T\mathcal{U}\boldsymbol{x}) - \frac{|\mathcal{U}\boldsymbol{x}|^2}{|\mathcal{V}\boldsymbol{x}|^4}\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^T\mathcal{V}^T\mathcal{V}\boldsymbol{x}) = \frac{2}{|\mathcal{V}(x)|^2}[\mathcal{U}^T\mathcal{U} - \frac{|\mathcal{U}\boldsymbol{x}|^2}{|\mathcal{V}\boldsymbol{x}|^2}\mathcal{V}\mathcal{V}^T]\boldsymbol{x},$$

therefore the minimum $\boldsymbol{x}_0$ of E must be a generalized eigenvector associated of the symmetric matrices $\mathcal{U}^T\mathcal{U}$ and $\mathcal{V}^T\mathcal{V}$.
Now consider any generalized eigenvector $\boldsymbol{x}$ of the symmetric matrices $\mathcal{U}^T\mathcal{U}$ and $\mathcal{V}^T\mathcal{V}$ and let λ denote the corresponding generalized eigenvalue. The value of E in $\boldsymbol{x}$ is obviously equal to λ. The generalized eigenvalue λ_0 associated with $\boldsymbol{x}_0$ is necessarily the smallest one since any smaller generalized eigenvalue would yield a smaller value for E. We can pick a vector $\boldsymbol{x}_0$ such that $|\mathcal{V}\boldsymbol{x}|^2 = 1$ as the solution of our problem since, as mentioned earlier, the value of E is obviously invariant under scale changes.

3.2. Show that the 2×2 matrix $\mathcal{U}^T\mathcal{U}$ involved in the line-fitting example from Section 3.1.1 is the matrix of second moments of inertia of the points p_i $(i = 1, \ldots, n)$.

Solution Recall that

$$\mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \ldots & \ldots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}.$$

Thus, omitting the summation indexes for conciseness, we have

$$
\begin{aligned}
\mathcal{U}^T\mathcal{U} &= \begin{pmatrix} \sum(x_i-\bar{x})^2 & \sum(x_i-\bar{x})(y_i-\bar{y}) \\ \sum(x_i-\bar{x})(y_i-\bar{y}) & \sum(y_i-\bar{y})^2 \end{pmatrix} \\
&= \begin{pmatrix} \sum(x_i^2-2\bar{x}x_i+\bar{x}^2) & \sum(x_iy_i-\bar{x}y_i-\bar{y}x_i+\bar{x}\bar{y}) \\ \sum(x_iy_i-\bar{x}y_i-\bar{y}x_i+\bar{x}\bar{y}) & \sum(y_i^2-2\bar{y}y_i+\bar{y}^2) \end{pmatrix} \\
&= \begin{pmatrix} \sum x_i^2-2\bar{x}\sum x_i+n\bar{x}^2 & \sum x_iy_i-\bar{x}\sum y_i-\bar{y}\sum x_i+n\bar{x}\bar{y} \\ \sum x_iy_i-\bar{x}\sum y_i-\bar{y}\sum x_i+n\bar{x}\bar{y} & \sum y_i^2-2\bar{y}\sum y_i+n\bar{y}^2 \end{pmatrix} \\
&= \begin{pmatrix} \sum x_i^2-n\bar{x}^2 & \sum x_iy_i-n\bar{x}\bar{y} \\ \sum x_iy_i-n\bar{x}\bar{y} & \sum y_i^2-n\bar{y}^2 \end{pmatrix},
\end{aligned}
$$

which is, indeed, the matrix of second moments of inertia of the points p_i.

3.3. Extend the line-fitting method presented in Section 3.1.1 to the problem of fitting a plane to n points in $\mathbb{E}^3$.

Solution We consider n points P_i $(i=1,\ldots,n)$ with coordinates $(x_i,y_i,z_i)^T$ in some fixed coordinate system, and find the plane with equation $ax+by+cz-d=0$ and unit normal $=\boldsymbol{n}=(a,b,c)^T$ that best fits these points. This amounts to minimizing

$$E(a,b,c,d)=\sum_{i=1}^{n}(ax_i+by_i+cz_i-d)^2$$

with respect to a, b, c, and d under the constraint $a^2+b^2+c^2=1$. Differentiating E with respect to d shows that, at a minimum of this function, we must have $0=\partial E/\partial d=-2\sum_{i=1}^{n}(ax_i+by_i+cz_i-d)$, thus

$$d=a\bar{x}+b\bar{y}+c\bar{z},\ \text{where}\ \bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i,\ \bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i,\ \text{and}\ \bar{z}=\frac{1}{n}\sum_{i=1}^{n}z_i. \tag{3.1}$$

Substituting this expression for d in the definition of E yields

$$E=\sum_{i=1}^{n}[a(x_i-\bar{x})+b(y_i-\bar{y})+c(z_i-\bar{z})]^2=|\mathcal{U}\boldsymbol{n}|^2,$$

where

$$\mathcal{U}=\begin{pmatrix} x_1-\bar{x} & y_1-\bar{y} & z_1-\bar{z} \\ \ldots & \ldots & \ldots \\ x_n-\bar{x} & y_n-\bar{y} & z_n-\bar{z} \end{pmatrix},$$

and our original problem finally reduces to minimizing $|\mathcal{U}\boldsymbol{n}|^2$ with respect to $\boldsymbol{n}$ under the constraint $|\boldsymbol{n}|^2=1$. We recognize a homogeneous linear least-squares problem, whose solution is the unit eigenvector associated with the minimum eigenvalue of the 3×3 matrix $\mathcal{U}^T\mathcal{U}$. Once a, b, and c have been computed, the value of d is immediately obtained from Eq. (3.1). Similar to the line-fitting case, we have

$$\mathcal{U}^T\mathcal{U}=\begin{pmatrix} \sum_{i=1}^{n}x_i^2-n\bar{x}^2 & \sum_{i=1}^{n}x_iy_i-n\bar{x}\bar{y} & \sum_{i=1}^{n}x_iz_i-n\bar{x}\bar{z} \\ \sum_{i=1}^{n}x_iy_i-n\bar{x}\bar{y} & \sum_{i=1}^{n}y_i^2-n\bar{y}^2 & \sum_{i=1}^{n}y_iz_i-n\bar{y}\bar{z} \\ \sum_{i=1}^{n}x_iz_i-n\bar{x}\bar{z} & \sum_{i=1}^{n}y_iz_i-n\bar{y}\bar{z} & \sum_{i=1}^{n}z_i^2-n\bar{y}^2 \end{pmatrix}$$

that is, the matrix of second moments of inertia of the points P_i.

3.4. Derive an expression for the Hessian of the functions $f_{2i-1}(\boldsymbol{\xi}) = \tilde{u}_i(\boldsymbol{\xi}) - u_i$ and $f_{2i}(\boldsymbol{\xi}) = \tilde{v}_i(\boldsymbol{\xi}) - v_i$ $(i = 1, \ldots, n)$ introduced in Section 3.4.

Solution Recall that the first-order partial derivatives of the components of f are given by:

$$\begin{pmatrix} \dfrac{\partial f_{2i-1}^T}{\partial \boldsymbol{\xi}} \\ \dfrac{\partial f_{2i}^T}{\partial \boldsymbol{\xi}} \end{pmatrix} = \frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \\ \mathbf{0}^T & \boldsymbol{P}_i^T & -\tilde{v}_i \boldsymbol{P}_i^T \end{pmatrix} \mathcal{J}\boldsymbol{m},$$

We just differentiate these expressions once more to get the Jacobian:

$$\begin{aligned}
\frac{\partial^2 f_{2i-1}^T}{\partial \xi_j \partial \xi_k} &= \frac{\partial}{\partial \xi_k}[\frac{1}{\tilde{z}_i}\begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial \boldsymbol{m}}{\partial \xi_j}] \\
&= \left[\frac{-1}{\tilde{z}_i^2} \frac{\partial \tilde{z}_i}{\partial \xi_k} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} + \frac{1}{\tilde{z}_i} \begin{pmatrix} \mathbf{0}^T & \mathbf{0}^T & -\frac{\partial \tilde{u}_i}{\partial \xi_k} \boldsymbol{P}_i^T \end{pmatrix} \right] \frac{\partial \boldsymbol{m}}{\partial \xi_j} \\
&+ \frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial^2 \boldsymbol{m}}{\partial \xi_j \partial \xi_k} \\
&= \left[\frac{-1}{\tilde{z}_i^2} (\frac{\partial \boldsymbol{m}_3}{\xi_k}^T \boldsymbol{P}_i) \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \right. \\
&\left. \quad + \frac{1}{\tilde{z}_i} \begin{pmatrix} \mathbf{0}^T & \mathbf{0}^T & -\frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial \boldsymbol{m}}{\partial \xi_k} \boldsymbol{P}_i^T \end{pmatrix} \right] \frac{\partial \boldsymbol{m}}{\partial \xi_j} \\
&+ \frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial^2 \boldsymbol{m}}{\partial \xi_j \partial \xi_k}.
\end{aligned}$$

Rearranging the terms finally yields

$$\frac{\partial^2 f_{2i-1}^T}{\partial \xi_j \partial \xi_k} = \frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial^2 \boldsymbol{m}}{\partial \xi_j \partial \xi_k} - \frac{1}{\tilde{z}_i} \frac{\partial \boldsymbol{m}}{\partial \xi_j}^T \begin{pmatrix} \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T & \boldsymbol{P}_i \boldsymbol{P}_i^T \\ \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T \\ \boldsymbol{P}_i \boldsymbol{P}_i^T & \mathbf{0}\mathbf{0}^T & -2\tilde{u}_i \boldsymbol{P}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial \boldsymbol{m}}{\partial \xi_k}.$$

The same line of reasoning can be used to show that

$$\frac{\partial^2 f_{2i}^T}{\partial \xi_j \partial \xi_k} = \frac{1}{\tilde{z}_i} \begin{pmatrix} \boldsymbol{P}_i^T & \mathbf{0}^T & -\tilde{u}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial^2 \boldsymbol{m}}{\partial \xi_j \partial \xi_k} - \frac{1}{\tilde{z}_i} \frac{\partial \boldsymbol{m}}{\partial \xi_j}^T \begin{pmatrix} \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T \\ \mathbf{0}\mathbf{0}^T & \mathbf{0}\mathbf{0}^T & \boldsymbol{P}_i \boldsymbol{P}_i^T \\ \mathbf{0}\mathbf{0}^T & \boldsymbol{P}_i \boldsymbol{P}_i^T & -2\tilde{v}_i \boldsymbol{P}_i \boldsymbol{P}_i^T \end{pmatrix} \frac{\partial \boldsymbol{m}}{\partial \xi_k}.$$

3.5. Euler angles. Show that the rotation obtained by first rotating about the z axis of some coordinate frame by an angle α, then rotating about the y axis of the new coordinate frame by an angle β and finally rotating about the z axis of the resulting frame by an angle γ can be represented in the original coordinate system by

$$\begin{pmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{pmatrix}.$$

Solution Let us denote by (A), (B), (C), and (D) the consecutive coordinate systems. If $\mathcal{R}_x(\theta)$, $\mathcal{R}_y(\theta)$, and $\mathcal{R}_z(\theta)$ denotes the rotation matrices about axes x,

y, and z, we have ${}^A_B\mathcal{R} = \mathcal{R}_z(\alpha)$, ${}^B_C\mathcal{R} = \mathcal{R}_y(\beta)$, and ${}^C_D\mathcal{R} = \mathcal{R}_z(\gamma)$. Thus

$$
\begin{aligned}
{}^A_D\mathcal{R} &= {}^A_B\mathcal{R}\,{}^B_C\mathcal{R}\,{}^C_D\mathcal{R}\\
&= \begin{pmatrix} \cos\alpha & -\sin\alpha & 0\\ \sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix} \cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{pmatrix}\begin{pmatrix} \cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{pmatrix}\\
&= \begin{pmatrix} \cos\alpha\cos\beta & -\sin\alpha & \cos\alpha\sin\beta\\ \sin\alpha\cos\beta & \cos\alpha & \sin\alpha\sin\beta\\ -\sin\beta & 0 & \cos\beta\end{pmatrix}\begin{pmatrix} \cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{pmatrix}\\
&= \begin{pmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta\end{pmatrix}.
\end{aligned}
$$

Now the rotation that maps (A) onto (D) maps a point P with position ${}^AP$ in the coordinate system (A) onto the point P' with the same position ${}^DP' = {}^AP$ *in the coordinate system* D. We have ${}^AP' = {}^A_D\mathcal{R}\,{}^DP' = {}^A_D\mathcal{R}\,{}^AP$, which proves the desired result.

3.6. The Rodrigues formula. Consider a rotation $\mathcal{R}$ of angle θ about the axis $\boldsymbol{u}$ (a unit vector). Show that $\mathcal{R}\boldsymbol{x} = \cos\theta\boldsymbol{x} + \sin\theta\,\boldsymbol{u}\times\boldsymbol{x} + (1-\cos\theta)(\boldsymbol{u}\cdot\boldsymbol{x})\boldsymbol{u}$.
Hint: A rotation does not change the projection of a vector $\boldsymbol{x}$ onto the direction $\boldsymbol{u}$ of its axis and applies a planar rotation of angle θ to the projection of $\boldsymbol{x}$ into the plane orthogonal to $\boldsymbol{u}$.

Solution Let $\boldsymbol{a}$ denote the orthogonal projection of $\boldsymbol{x}$ onto $\boldsymbol{u}$, $\boldsymbol{b} = \boldsymbol{x} - \boldsymbol{a}$ denote its orthogonal projection onto the plane perpendicular to $\boldsymbol{u}$, and $\boldsymbol{c} = \boldsymbol{u}\times\boldsymbol{b}$. By construction $\boldsymbol{c}$ is perpendicular to both $\boldsymbol{u}$ and $\boldsymbol{b}$, and, according to the property of rotation matrices mentioned in the hint, we must have $\mathcal{R}\boldsymbol{x} = \boldsymbol{a} + \cos\theta\boldsymbol{b} + \sin\theta\boldsymbol{c}$. Obviously, we also have

$$
\begin{cases}
\boldsymbol{a} = (\boldsymbol{u}\cdot\boldsymbol{x})\boldsymbol{u},\\
\boldsymbol{b} = \boldsymbol{x} - (\boldsymbol{u}\cdot\boldsymbol{x})\boldsymbol{u},\\
\boldsymbol{c} = \boldsymbol{u}\times\boldsymbol{x},
\end{cases}
$$

and it follows that $\mathcal{R}\boldsymbol{x} = \cos\theta\boldsymbol{x} + \sin\theta\,\boldsymbol{u}\times\boldsymbol{x} + (1-\cos\theta)(\boldsymbol{u}\cdot\boldsymbol{x})\boldsymbol{u}$.

3.7. Use the Rodrigues formula to show that the matrix $\mathcal{R}$ associated with a rotation of angle θ about the unit vector $\boldsymbol{u} = (u, v, w)^T$

$$
\begin{pmatrix} u^2(1-c)+c & uv(1-c)-ws & uw(1-c)+vs\\ uv(1-c)+ws & v^2(1-c)+c & vw(1-c)-us\\ uw(1-c)-vs & vw(1-c)+us & w^2(1-c)+c\end{pmatrix}.
$$

where $c = \cos\theta$ and $s = \sin\theta$.

Solution With the notation $\boldsymbol{u} = (u, v, w)^T$, $c = \cos\theta$ and $s = \sin\theta$, the Rodrigues formula is easily rewritten as

$$
\mathcal{R}\boldsymbol{x} = \left(c\mathrm{Id} + s[\boldsymbol{u}_\times] + (1-c)\boldsymbol{u}\boldsymbol{u}^T\right)\boldsymbol{x},
$$

and it follows that

$$
\mathcal{R} = c\begin{pmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} + s\begin{pmatrix} 0 & -w & v\\ w & 0 & -u\\ -v & u & 0\end{pmatrix} + (1-c)\begin{pmatrix} u^2 & uv & uw\\ uv & v^2 & vw\\ uw & vw & w^2\end{pmatrix},
$$

and the result follows immediately.

3.8. Assuming that the intrinsic parameters of a camera are known, show how to compute its extrinsic parameters once the vector $\boldsymbol{n}'$ defined in Section 3.5 is known. Hint: Use the fact that the rows of a rotation matrix form an orthonormal family.

Solution Recall that the vector $\boldsymbol{n}' = (m_{11}, m_{12}, m_{14}, m_{21}, m_{22}, m_{24})^T$ can only be recovered up to scale. With the intrinsic parameters known, this means that we can write the projection matrix as $\mathcal{M} = \begin{pmatrix} \mathcal{A} & \boldsymbol{b} \end{pmatrix} = \rho \begin{pmatrix} \mathcal{R} & \boldsymbol{t} \end{pmatrix}$, where $\mathcal{R}$ and $\boldsymbol{t}$ are the rotation matrix and translation vector associated with the camera's extrinsic parameters.
Let $\boldsymbol{a}_1^T$ and $\boldsymbol{a}_2^T$ denote as usual the two rows of the matrix $\mathcal{A}$. Since the rows of a rotation matrix have unit norm and are orthogonal to each other, we have $|\boldsymbol{a}_1^2| = |\boldsymbol{a}_2^2| = \rho^2$ and $\boldsymbol{a}_1 \cdot \boldsymbol{a}_2 = 0$. These two constraints can be seen as quadratic equations in the unknowns m_{13} and m_{23}, namely

$$\begin{cases} m_{23}^2 - m_{13}^2 = |\boldsymbol{b}_1|^2 - |\boldsymbol{b}_2|^2, \\ m_{13}m_{23} = -\boldsymbol{b}_1 \cdot \boldsymbol{b}_2, \end{cases}$$

where $\boldsymbol{b}_1 = (m_{11}, m_{12})^T$ and $\boldsymbol{b}_2 = (m_{21}, m_{22})^T$. Squaring the second equation and substituting the value of m_{23}^2 from the first equation into it yields

$$m_{13}^2[m_{13}^2 + |\boldsymbol{b}_1|^2 - |\boldsymbol{b}_2|^2] = (\boldsymbol{b}_1 \cdot \boldsymbol{b}_2)^2,$$

or equivalently

$$m_{13}^4 + (|\boldsymbol{b}_1|^2 - |\boldsymbol{b}_2|^2)m_{13}^2 - (\boldsymbol{b}_1 \cdot \boldsymbol{b}_2)^2 = 0.$$

This is a quadratic equation in m_{13}^2. Since the constant term and the quadratic term have opposite signs, it always admits two real solutions with opposite signs. Only the positive one is valid of course, and it yields two opposite solutions for m_{13}. The remaining unknown is then determined as $m_{23} = -(\boldsymbol{b}_1 \cdot \boldsymbol{b}_2)/m_{13}$.
At this point, there are four valid values for the triple $(\boldsymbol{a}_1, \boldsymbol{a}_2, \rho)$ since m_{13} and m_{23} are determined up to a single sign ambiguity, and the value of ρ is determined up to a second sign ambiguity by $\rho^2 = |\boldsymbol{a}_1|^2$. In turn, this determines four valid values for the rows $\boldsymbol{r}_1^T$ and $\boldsymbol{r}_2^T$ of $\mathcal{R}$ and the coordinates t_x and t_y of $\boldsymbol{t}$. For each of these solutions, the last row of $\mathcal{R}$ is computed as $\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$, which gives in turn $\boldsymbol{a}_3 = \rho \boldsymbol{r}_3$. Finally, an initial value of $t_z = m_{14}/\rho$ can be computed using linear least squares by setting $\lambda = 1$ in Eq. (3.23). The correct solution among the four found can be identified by (a) using the sign of t_z (when it is known) to discard obviously incorrect solutions, and (b) picking among the remaining ones the solution that yields the smallest residual in the least-squares estimation process.

3.9. Assume that n fiducial lines with known Plücker coordinates are observed by a camera.

(a) Show that the line projection matrix $\tilde{\mathcal{M}}$ introduced in the exercises of chapter 2 can be recovered using linear least squares when $n \geq 9$.

(b) Show that once $\tilde{\mathcal{M}}$ is known, the projection matrix $\mathcal{M}$ can also be recovered using linear least squares.

Hint: Consider the rows $\boldsymbol{m}_i$ of $\mathcal{M}$ as the coordinate vectors of three planes Π_i and the rows $\tilde{\boldsymbol{m}}_i$ of $\tilde{\mathcal{M}}$ as the coordinate vectors of three lines, and use the incidence relationships between these planes and these lines to derive linear constraints on the vectors $\boldsymbol{m}_i$.

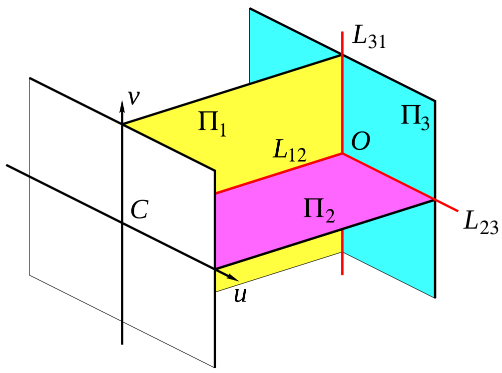
Solution

(a) We saw in Exercise 2.15 that the Plücker coordinate vector of a line Δ and the homogeneous coordinate vector of its image δ are related by

$$\rho\boldsymbol{\delta} = \tilde{\mathcal{M}}\boldsymbol{\Delta}, \quad \text{where} \quad \tilde{\mathcal{M}} \stackrel{\text{def}}{=} \begin{pmatrix} (\boldsymbol{m}_2 \wedge \boldsymbol{m}_3)^T \\ (\boldsymbol{m}_3 \wedge \boldsymbol{m}_1)^T \\ (\boldsymbol{m}_1 \wedge \boldsymbol{m}_2)^T \end{pmatrix}.$$

We can eliminate the unknown scale factor ρ by using the fact that the cross product of two parallel vectors is zero, thus $\boldsymbol{\delta} \times \tilde{\mathcal{M}}\boldsymbol{\Delta} = \mathbf{0}$. This linear vector equation in the components of $\tilde{\mathcal{M}}$ is equivalent to two independent scalar equations. Since the 3×6 matrix $\tilde{\mathcal{M}}$ is only defined up to scale, its 17 independent coefficients can thus be estimated as before via linear least squares (ignoring the non-linear constraints imposed by the fact that the rows of $\tilde{\mathcal{M}}$ are Plücker coordinate vectors) when $n \geq 9$.

(b) Once $\tilde{\mathcal{M}}$ is known, we can recover $\mathcal{M}$ as well through linear least squares. Indeed, the vectors $\boldsymbol{m}_i$ $(i = 1, 2, 3)$ can be thought of as the homogeneous coordinate vectors of three projection planes Π_i (see diagram below). These planes intersect at the optical center O of the camera since the homogeneous coordinate vector of this point satisfies the equation $\mathcal{M}\boldsymbol{O} = \mathbf{0}$. Likewise, it is easy to show that Π_3 is parallel to the image plane, that Π_3 and Π_1 intersect along a line L_{31} parallel to the $u = 0$ coordinate axis of the image plane, that Π_2 and Π_3 intersect along a line L_{23} parallel to its $v = 0$ coordinate axis, and that the line L_{12} formed by the intersection of Π_1 and Π_2 is simply the optical axis.



According to Exercise 2.15, if $\boldsymbol{L}_{12} = \boldsymbol{m}_1 \wedge \boldsymbol{m}_2$, we can write that L_{12} lies in the plane Π_1 as $\mathcal{L}_{12}\boldsymbol{m}_1 = 0$, where $\mathcal{L}_{12}$ is the 4×4 matrix associated with the vector $\boldsymbol{L}_{12}$. Five more homogeneous constraints on the vectors $\boldsymbol{m}_i$ $(i = 1, 2, 3)$ are obtained by appropriate permutations of the indices. In addition, we must have

$$\rho_{12}\mathcal{L}_{12}\boldsymbol{m}_3 = \rho_{23}\mathcal{L}_{23}\boldsymbol{m}_1 = \rho_{31}\mathcal{L}_{31}\boldsymbol{m}_2$$

for some non-zero scalars ρ_{12}, ρ_{23} and ρ_{31} since each matrix product in this equation gives the homogeneous coordinate vector of the optical point. In fact, it is easy to show that the three scale factors can be taken equal to each other, which yields three homogeneous vector equations in $\boldsymbol{m}_i$ $(i = 1, 2, 3)$:

$$\mathcal{L}_{12}\boldsymbol{m}_3 = \mathcal{L}_{23}\boldsymbol{m}_1 = \mathcal{L}_{31}\boldsymbol{m}_2.$$

Putting it all together, we obtain $6 \times 3 + 3 \times 4 = 30$ homogeneous (scalar) linear equations in the coefficients of $\mathcal{M}$, whose solution can once again be found via linear least squares (at most 11 of the 30 equations are independent in the noise-free case). Once $\mathcal{M}$ is known, the intrinsic and extrinsic parameters can be computed as before. We leave to the reader the task of characterizing the degenerate line configurations for which the proposed method fails.

Programming Assignments

3.10. Use linear least-squares to fit a plane to n points $(x_i, y_i, z_i)^T$ $(i = 1, \ldots, n)$ in $\mathbb{R}^3$.

3.11. Use linear least-squares to fit a conic section defined by $ax^2+bxy+cy^2+dx+ey+f=0$ to n points $(x_i, y_i)^T$ $(i = 1, \ldots, n)$ in $\mathbb{R}^2$.

3.12. Implement the linear calibration algorithm presented in Section 3.2.

3.13. Implement the calibration algorithm that takes into account radial distortion and that was presented in Section 3.3.

3.14. Implement the nonlinear calibration algorithm from Section 3.4.

C H A P T E R 4

Radiometry—Measuring Light

PROBLEMS

4.1. How many steradians in a hemisphere?

Solution 2π.

4.2. We have proved that radiance does not go down along a straight line *in a non-absorbing medium*, which makes it a useful unit. Show that if we were to use power per square meter of foreshortened area (which is irradiance), the unit must change with distance along a straight line. How significant is this difference?

Solution Assume we have a source and two receivers that look exactly the same from the source. One is large and far away, the other is small and nearby. Because they look the same from the source, exactly the same rays leaving the source pass through each receiver. If our unit is power per square meter of foreshortened area, the amount of power arriving at a receiver is given by integrating this over the area of the receiver. But the distant one is bigger, and so if the value of power per square meter of foreshortened area didn't go down with distance, then it would receive more power than the nearby receiver, which is impossible (how does the source know which one should get more power?).

4.3. **An absorbing medium:** Assume that the world is filled with an isotropic absorbing medium. A good, simple model of such a medium is obtained by considering a line along which radiance travels. If the radiance along the line is N at x, it is $N - (\alpha dx)N$ at $x + dx$.

(a) Write an expression for the radiance transferred from one surface patch to another in the presence of this medium.

(b) Now *qualitatively* describe the distribution of light in a room filled with this medium for α small and large positive numbers. The room is a cube, and the light is a single small patch in the center of the ceiling. Keep in mind that if α is large and positive, little light actually reaches the walls of the room.

Solution

(a) $\frac{dN}{dx} = -\alpha N$.

(b) Radiance goes down exponentially with distance. Assume the largest distance in the room is d. If α is small enough — much less than $1/d$, room looks like usual. As α gets bigger, interreflections are quenched; for large α only objects that view the light directly and are close to the light will be bright.

4.4. Identify common surfaces that are neither Lambertian nor specular using the underside of a CD as a working example. There are a variety of important biological examples, which are often blue in color. Give at least two different reasons that it could be advantageous to an organism to have a non-Lambertian surface.

Solution There are lots. Many possible advantages; for example, an animal that looks small to a predator approaching in one direction (because it looks dark from this direction) could turn quickly and look big (because it looks bright from this direction).

4.5. Show that for an ideal diffuse surface the directional hemispheric reflectance is constant; now show that if a surface has constant directional hemispheric reflectance, it is ideal diffuse.

Solution In an ideal diffuse surface, the BRDF is constant; DHR is an integral of the BRDF over the outgoing angles, and so must be constant too. The other direction is false (sorry!—DAF).

4.6. Show that the BRDF of an ideal specular surface is

$$\rho_{bd}(\theta_o, \phi_o, \theta_i, \phi_i) = \rho_s(\theta_i)\{2\delta(\sin^2\theta_o - \sin^2\theta_i)\}\{\delta(\phi_o - \phi\pi)\},$$

where $\rho_s(\theta_i)$ is the fraction of radiation that leaves.

Solution See the book, *Radiosity and Global Illumination*, F. Sillion and C. Puech, Morgan-Kauffman, 1994 — this is worked on page 16.

4.7. Why are specularities brighter than diffuse reflection?

Solution Because the reflected light is concentrated in a smaller range of angles.

4.8. A surface has constant BRDF. What is the maximum possible value of this constant? Now assume that the surface is known to absorb 20% of the radiation incident on it (the rest is reflected); what is the value of the BRDF?

Solution $1/\pi$; $0.8/\pi$.

4.9. The eye responds to radiance. Explain why Lambertian surfaces are often referred to as having a brightness independent of viewing angle.

Solution The radiance leaving an ideal diffuse surface is independent of exit angle.

4.10. Show that the solid angle subtended by a sphere of radius ϵ at a point a distance r away from the center of the sphere is approximately $\pi(\frac{\epsilon}{r})^2$, for $r \gg \epsilon$.

Solution If the sphere is far enough, then the rays from the sphere to the point are approximately parallel, and the cone of rays intersects the sphere in a circle of radius ϵ. This circle points toward the point, and the rest is simple calculation.

CHAPTER 5

Sources, Shadows and Shading

PROBLEMS

5.1. What shapes can the shadow of a sphere take if it is cast on a plane and the source is a point source?

Solution Any conic section. The vertex of the cone is the point source; the rays tangent to the sphere form a right circular cone, and this cone is sliced by a plane. It's not possible to get both parts of a hyperbola.

5.2. We have a square area source and a square occluder, both parallel to a plane. The source is the same size as the occluder, and they are vertically above one another with their centers aligned.
(a) What is the shape of the umbra?
(b) What is the shape of the outside boundary of the penumbra?

Solution

(a) Square.

(b) Construct with a drawing, to get an eight-sided polygon.

5.3. We have a square area source and a square occluder, both parallel to a plane. The edge length of the source is now twice that of the occluder, and they are vertically above one another with their centers aligned.
(a) What is the shape of the umbra?
(b) What is the shape of the outside boundary of the penumbra?

Solution

(a) Depends how far the source is above the occluder; it could be absent, or square.

(b) Construct with a drawing, to get an eight sided polygon.

5.4. We have a square area source and a square occluder, both parallel to a plane. The edge length of the source is now half that of the occluder, and they are vertically above one another with their centers aligned.
(a) What is the shape of the umbra?
(b) What is the shape of the outside boundary of the penumbra?

Solution (a) Square. (b) Construct with a drawing, to get an eight sided polygon.

5.5. A small sphere casts a shadow on a larger sphere. Describe the possible shadow boundaries that occur.

Solution Very complex, given by the intersection of a right circular cone and a sphere. In the simplest case, the two centers are aligned with the point source, and the shadow is a circle.

5.6. Explain why it is difficult to use shadow boundaries to infer shape, particularly if the shadow is cast onto a curved surface.

Solution As the example above suggests, the boundary is a complicated 3D curve. Typically, we see this curve projected into an image. This means it is very difficult to infer anything except in quite special cases (e.g. projection of the shadow boundary onto a plane).

5.7. An infinitesimal patch views a circular area source of constant exitance frontally along the axis of symmetry of the source. Compute the radiosity of the patch due to the source exitance $E(\boldsymbol{u})$ as a function of the area of the source and the distance between the center of the source and the patch. You may have to look the integral up in tables — if you don't, you're entitled to feel pleased with yourself — but this is one of few cases that can be done in closed form. It is easier to look up if you transform it to get rid of the cosine terms.

Solution Radiosity is proportional to $A/(A + \pi d^2)$, where A is the area of the source and d is the distance.

5.8. As in Figure 5.17, a small patch views an infinite plane at unit distance. The patch is sufficiently small that it reflects a trivial quantity of light onto the plane. The plane has radiosity $B(x, y) = 1 + \sin ax$. The patch and the plane are parallel to one another. We move the patch around parallel to the plane, and consider its radiosity at various points.

(a) Show that if one translates the patch, its radiosity varies periodically with its position in x.

(b) Fix the patch's center at $(0, 0)$; determine a *closed form* expression for the radiosity of the patch at this point as a function of a. You'll need a table of integrals for this (if you do not, you are entitled to feel very pleased with yourself).

Solution

(a) Obvious symmetry in the geometry.

(b) This integral is in "Shading primitives", J. Haddon and D.A. Forsyth, Proc. Int. Conf. Computer Vision, 1997.

5.9. If one looks across a large bay in the daytime, it is often hard to distinguish the mountains on the opposite side; near sunset, they are clearly visible. This phenomenon has to do with scattering of light by air — a large volume of air is actually a source. Explain what is happening. We have modeled air as a vacuum and asserted that no energy is lost along a straight line in a vacuum. Use your explanation to give an estimate of the kind of scales over which that model is acceptable.

Solution In the day, the air between you and the other side is illuminated by the sun; some light scatters toward your eye. This has the effect of reducing contrast, meaning that the other side of the bay is hard to see because it's about as bright as the air. By evening, the air is less strongly illuminated and the contrast goes up. This suggests that assuming air doesn't interact with light is probably dubious at scales of multiple kilometers (considerably less close to a city).

5.10. Read the book *Colour and Light in Nature*, by Lynch and Livingstone, published by Cambridge University Press, 1995.

Programming Assignments

5.11. An area source can be approximated as a grid of point sources. The weakness of this approximation is that the penumbra contains quantization errors, which can be quite offensive to the eye.

(a) Explain.

(b) Render this effect for a square source and a single occluder casting a shadow onto an infinite plane. For a fixed geometry, you should find that as the number of point sources goes up, the quantization error goes down.

(c) This approximation has the unpleasant property that it is possible to produce arbitrarily large quantization errors with any finite grid by changing the geometry. This is because there are configurations of source and occluder that produce large penumbrae. Use a square source and a single occluder, casting a shadow onto an infinite plane, to explain this effect.

5.12. Make a world of black objects and another of white objects (paper, glue and spray-paint are useful here) and observe the effects of interreflections. Can you come up with a criterion that reliably tells, *from an image*, which is which? (If you can, publish it; the problem looks easy, but isn't).

5.13. (This exercise requires some knowledge of numerical analysis.) Do the numerical integrals required to reproduce Figure 5.17. These integrals aren't particularly easy: If one uses coordinates on the infinite plane, the size of the domain is a nuisance; if one converts to coordinates on the view hemisphere of the patch, the frequency of the radiance becomes infinite at the boundary of the hemisphere. The best way to estimate these integrals is using a Monte Carlo method on the hemisphere. You should use importance sampling because the boundary contributes rather less to the integral than the top.

5.14. Set up and solve the linear equations for an interreflection solution for the interior of a cube with a small square source in the center of the ceiling.

5.15. Implement a photometric stereo system.

(a) How accurate are its measurements (i.e., how well do they compare with known shape information)? Do interreflections affect the accuracy?

(b) How repeatable are its measurements (i.e., if you obtain another set of images, perhaps under different illuminants, and recover shape from those, how does the new shape compare with the old)?

(c) Compare the minimization approach to reconstruction with the integration approach; which is more accurate or more repeatable and why? Does this difference appear in experiment?

(d) One possible way to improve the integration approach is to obtain depths by integrating over many different paths and then average these depths (you need to be a little careful about constants here). Does this improve the accuracy or repeatability of the method?

CHAPTER 6

Color

PROBLEMS

6.1. Sit down with a friend and a packet of colored papers, and compare the color names that you use. You need a large packet of papers — one can very often get collections of colored swatches for paint, or for the Pantone color system very cheaply. The best names to try are basic color names — the terms *red*, *pink*, *orange*, *yellow*, *green*, *blue*, *purple*, *brown*, *white*, *gray* and *black*, which (with a small number of other terms) have remarkable canonical properties that apply widely across different languages (the papers in **?**) give a good summary of current thought on this issue). You will find it surprisingly easy to disagree on which colors should be called blue and which green, for example.

Solution Students should do the experiment; there's no right answer, but if two people agree on all color names for all papers with a large range of colour, something funny is going on.

6.2. Derive the equations for transforming from RGB to CIE XYZ and back. This is a linear transformation. It is sufficient to write out the expressions for the elements of the linear transformation — you don't have to look up the actual numerical values of the color matching functions.

Solution Write the RGB primaries as $p_r(\lambda)$, $p_g(\lambda)$, $p_b(\lambda)$. If a colour has RGB coords. (a, b, c), that means it matches $ap_r(\lambda)+bp_g(\lambda)+cp_b(\lambda)$. What are the XYZ coord.'s (d, e, f) of this colour? we compute them with the XYZ colour matching functions $x(\lambda)$, $y(\lambda)$ and $z(\lambda)$, to get

$$\begin{pmatrix} d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \int x(\lambda)p_r(\lambda)d\lambda & \int x(\lambda)p_g(\lambda)d\lambda & \int x(\lambda)p_b(\lambda)d\lambda \\ \int y(\lambda)p_r(\lambda)d\lambda & \int y(\lambda)p_g(\lambda)d\lambda & \int y(\lambda)p_b(\lambda)d\lambda \\ \int z(\lambda)p_r(\lambda)d\lambda & \int z(\lambda)p_g(\lambda)d\lambda & \int z(\lambda)p_b(\lambda)d\lambda \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}.$$

6.3. Linear color spaces are obtained by choosing primaries and then constructing color matching functions for those primaries. Show that there is a linear transformation that takes the coordinates of a color in one linear color space to those in another; the easiest way to do this is to write out the transformation in terms of the color matching functions.

Solution Look at the previous answer.

6.4. Exercise 6.3 means that, in setting up a linear color space, it is possible to choose primaries arbitrarily, but there are constraints on the choice of color matching functions. Why? What are these constraints?

Solution Assume I have some linear color space and know its color matching functions. Then the color matching functions for any other linear color space are a linear combination of the color matching functions for this space. Arbitrary functions don't have this property, so we can't choose color matching functions arbitrarily.

6.5. Two surfaces that have the same color under one light and different colors under another are often referred to as *metamers.* An *optimal color* is a spectral reflectance or radiance that has value 0 at some wavelengths and 1 at others. Although optimal colors don't occur in practice, they are a useful device (due to Ostwald) for explaining various effects.

(a) Use optimal colors to explain how metamerism occurs.

(b) Given a particular spectral albedo, show that there are an infinite number of metameric spectral albedoes.

(c) Use optimal colors to construct an example of surfaces that look different under one light (say, red and green) and the same under another.

(d) Use optimal colors to construct an example of surfaces that swop apparent color when the light is changed (i.e., surface one looks red and surface two looks green under light one, and surface one looks green and surface two looks red under light two).

Solution

(a) See Figure 6.1.

(b) You can either do this graphically by extending the reasoning of Figure 6.1, or analytically.

(c,d) This follows directly from (a) and (b).

6.6. You have to map the gamut for a printer to that of a monitor. There are colors in each gamut that do not appear in the other. Given a monitor color that can't be reproduced exactly, you could choose the printer color that is closest. Why is this a bad idea for reproducing images? Would it work for reproducing "business graphics" (bar charts, pie charts, and the like, which all consist of many differernt large blocks of a single color)?

Solution Some regions that, in the original picture had smooth gradients will now have a constant color. Yes.

6.7. *Volume color* is a phenomenon associated with translucent materials that are colored — the most attractive example is a glass of wine. The coloring comes from different absorption coefficients at different wavelengths. Explain (a) why a small glass of sufficiently deeply colored red wine (a good Cahors or Gigondas) looks black (b) why a big glass of lightly colored red wine also looks black. Experimental work is optional.

Solution Absorption is exponential with distance; if the rate of absorption is sufficiently large for each wavelength, within a relatively short distance all will be absorbed and the wine will look black. If the coefficients are smaller, you need a larger glass to get the light absorbed.

6.8. (This exercise requires some knowledge of numerical analysis.) In Section 6.5.2, we set up the problem of recovering the log albedo for a set of surfaces as one of minimizing

$$| \mathcal{M}_x \boldsymbol{l} - \boldsymbol{p} |^2 + | \mathcal{M}_y \boldsymbol{l} - \boldsymbol{q} |^2,$$

where $\mathcal{M}_x$ forms the x derivative of $\boldsymbol{l}$ and $\mathcal{M}_y$ forms the y derivative (i.e., $\mathcal{M}_x \boldsymbol{l}$ is the x-derivative).

(a) We asserted that $\mathcal{M}_x$ and $\mathcal{M}_y$ existed. Use the expression for forward differences (or central differences, or any other difference approximation to the derivative) to form these matrices. Almost every element is zero.

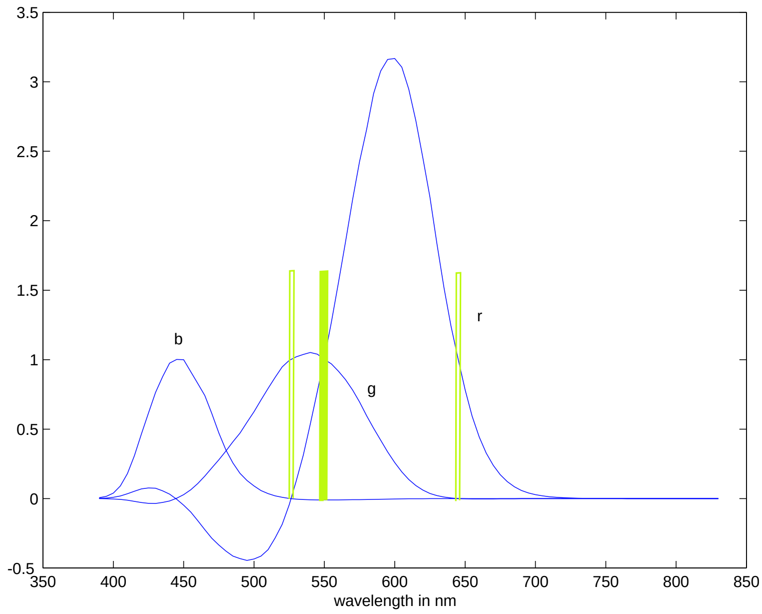


FIGURE 6.1: *The figure shows the RGB colormatching functions. The reflectance given by the two narrow bars is metameric to that given by the single, slightly thicker bar under uniform illumination, because under uniform illumination either reflectance will cause no response in B and about the same response in R and G. However, if the illuminant has high energy at about the center wavelength of the thicker bar and no energy elsewhere, the surface with this reflectance will look the same as it does under a uniform illuminant but the other one will be dark. It's worth trying to do a few other examples* with this sort of graphical reasoning *because it will give you a more visceral sense of what is going on than mere algebraic manipulation.*

(b) The minimization problem can be written in the form

$$\text{choose } \boldsymbol{l} \text{ to minimize } (\mathcal{A}\boldsymbol{l}+\boldsymbol{b})^T(\mathcal{A}\boldsymbol{l}+b).$$

Determine the values of $\mathcal{A}$ and $\boldsymbol{b}$, and show how to solve this general problem. You will need to keep in mind that $\mathcal{A}$ does not have full rank, so you can't go inverting it.

Solution

(a) Straightforward detail.

(b) The difficulty is the constant of integration. The problem is

$$\text{choose } \boldsymbol{l} \text{ to minimize}(\mathcal{A}\boldsymbol{l}+\boldsymbol{b})^T(\mathcal{A}\boldsymbol{l}+\boldsymbol{b})$$

or, equivalently,

$$\text{choose } \boldsymbol{l} \text{ so that } (\mathcal{A}^T\mathcal{A})\boldsymbol{l} = -\mathcal{A}^T\boldsymbol{b},$$

(which is guaranteed to have at least one solution). Actually, it must have at least a 2D space of solutions; one chooses an element of this space such that (say) the sum of values is a constant.

6.9. In Section 6.5.2, we mentioned two assumptions that would yield a constant of integration.

(a) Show how to use these assumptions to recover an albedo map.

(b) For each assumption, describe a situation where it fails, and describe the nature of the failure. Your examples should work for cases where there are many different albedoes in view.

Solution Assuming the brightest patch is white yields an albedo map because we know the difference of the logs of albedos for every pair of albedos; so we can set the brightest patch to have the value 1, and recover all others from these known ratios. Assuming the average lightness is a fixed constant is equally easy — if we choose some patches albedo to be an unknown constant c, each other albedo will be some known factor times c, so the average lightness will be some known term (a spatial average of the factors) times c. But this is assumed to be a known constant, so we get c. If the lightest albedo in view is not white, everything will be reported as being lighter than it actually is. If the spatial average albedo is different from the constant it is assumed to be, everything will be lighter or darker by the ratio of the constant to the actual vallue.

6.10. Read the book *Colour: Art and Science*, by Lamb and Bourriau, Cambridge University Press, 1995.

Programming Assignments

6.11. Spectra for illuminants and for surfaces are available on the web (try `http://www.it.lut.fi/research/color/lutcs_database.html`). Fit a finite-dimensional linear model to a set of illuminants and surface reflectances using principal components analysis, render the resulting models, and compare your rendering with an exact rendering. Where do you get the most significant errors? Why?

6.12. Print a colored image on a color inkjet printer using different papers and compare the result. It is particularly informative to (a) ensure that the driver knows what paper the printer will be printing on, and compare the variations in colors (which are ideally imperceptible), and (b) deceive the driver about what paper it is printing on (i.e., print on plain paper and tell the driver it is printing on photographic paper). Can you explain the variations you see? Why is photographic paper glossy?

6.13. Fitting a finite-dimensional linear model to illuminants and reflectances separately is somewhat ill-advised because there is no guarantee that the *interactions* will be represented well (they're not accounted for in the fitting error). It turns out that one can obtain g_{ijk} by a fitting process that sidesteps the use of basis functions. Implement this procedure (which is described in detail in **?**)), and compare the results with those obtained from the previous assignment.

6.14. Build a color constancy algorithm that uses the assumption that the spatial average of reflectance is constant. Use finite-dimensional linear models. You can get values of g_{ijk} from your solution to Exercise 3.

6.15. We ignore color interreflections in our surface color model. Do an experiment to get some idea of the size of color shifts possible from color interreflections (which are astonishingly big). Humans seldom interpret color interreflections as surface color. Speculate as to why this might be the case, using the discussion of the lightness algorithm as a guide.

6.16. Build a specularity finder along the lines described in Section 6.4.3

CHAPTER 7

Linear Filters

PROBLEMS

7.1. Show that forming unweighted local averages, which yields an operation of the form

$$\mathcal{R}_{ij} = \frac{1}{(2k+1)^2} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} \mathcal{F}_{uv}$$

is a convolution. What is the kernel of this convolution?

Solution Get this by pattern matching between formulae.

7.2. Write $\mathcal{E}_0$ for an image that consists of all zeros with a single one at the center. Show that convolving this image with the kernel

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{((i-k-1)^2 + (j-k-1)^2)}{2\sigma^2}\right)$$

(which is a discretised Gaussian) yields a circularly symmetric fuzzy blob.

Solution Convolving this image with any kernel reproduces the kernel; the current kernel is a circularly symmetric fuzzy blob.

7.3. Show that convolving an image with a discrete, separable 2D filter kernel is equivalent to convolving with two 1D filter kernels. Estimate the number of operations saved for an NxN image and a $2k+1 \times 2k+1$ kernel.

Solution

7.4. Show that convolving a function with a δ function simply reproduces the original function. Now show that convolving a function with a shifted δ function shifts the function.

Solution Simple index jockeying.

7.5. We said that convolving the image with a kernel of the form $(\sin x \sin y)/(xy)$ is impossible because this function has infinite support. Why would it be impossible to Fourier transform the image, multiply the Fourier transform by a box function, and then inverse-Fourier transform the result? Hint: Think support.

Solution You'd need an infinite image.

7.6. Aliasing takes high spatial frequencies to low spatial frequencies. Explain why the following effects occur:

(a) In old cowboy films that show wagons moving, the wheel often seems to be stationary or moving in the wrong direction (i.e., the wagon moves from left to right and the wheel seems to be turning counterclockwise).

(b) White shirts with thin dark pinstripes often generate a shimmering array of colors on television.

(c) In ray-traced pictures, soft shadows generated by area sources look blocky.

Solution

(a) The wheel has a symmetry, and has rotated just enough to look like itself, and so is stationary. It moves the wrong way when it rotates just too little to look like itself.

(b) Typically, color images are obtained by using three different sites on the same imaging grid, each sensitive to a different range of wavelengths. If the blue site in the camera sees on a stripe and the nearby red and green sites see on the shirt, the pixel reports yellow; but a small movement may mean (say) the green sees the stripe and the red and blue see the shirt, and we get purple.

(c) The source has been subdivided into a grid with point sources at the vertices; each block boundary occurs when one of these elements disappears behind, or reappears from behind, an occluder.

Programming Assignments

7.7. One way to obtain a Gaussian kernel is to convolve a constant kernel with itself many times. Compare this strategy with evaluating a Gaussian kernel.

(a) How many repeated convolutions do you need to get a reasonable approximation? (You need to establish what a reasonable approximation is; you might plot the quality of the approximation against the number of repeated convolutions).

(b) Are there any benefits that can be obtained like this? (Hint: Not every computer comes with an FPU.)

7.8. Write a program that produces a Gaussian pyramid from an image.

7.9. A sampled Gaussian kernel must alias because the kernel contains components at arbitrarily high spatial frequencies. Assume that the kernel is sampled on an infinite grid. As the standard deviation gets smaller, the aliased energy must increase. Plot the energy that aliases against the standard deviation of the Gaussian kernel in pixels. Now assume that the Gaussian kernel is given on a 7x7 grid. If the aliased energy must be of the same order of magnitude as the error due to truncating the Gaussian, what is the smallest standard deviation that can be expressed on this grid?

CHAPTER 8

Edge Detection

PROBLEMS

8.1. Each pixel value in 500×500 pixel image $\mathcal{I}$ is an independent, normally distributed random variable with zero mean and standard deviation one. Estimate the number of pixels that, where the absolute value of the x derivative, estimated by forward differences (i.e., $|I_{i+1,j} - I_{i,j}|$, is greater than 3.

Solution The signed difference has mean 0 and standard deviation $\sqrt{2}$. There are 500 rows and 499 differences per row, so a total of 500×499 differences. The probability that the absolute value of a difference is larger than 3 is

$$P(\text{diff} > 3) = \int_3^{\infty} \frac{1}{\sqrt{2}\sqrt{2\pi}} e^{\left(-x^2/4\right)} dx + \int_{-3}^{-\infty} \frac{1}{\sqrt{2}\sqrt{2\pi}} e^{\left(-x^2/4\right)} dx$$

and the answer is $500 * 499 * P(\text{diff} > 3)$.
$P(\text{diff} > 3)$ can be obtained from tables for the *complementary error function*, defined by

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-u^2} du.$$

Notice that

$$\frac{1}{\sqrt{2\pi}\sigma} \int_a^{\infty} e^{-\frac{u^2}{2\sigma^2}} du = \frac{1}{\sqrt{\pi}} \int_{\frac{a}{\sqrt{2}\sigma}}^{\infty} e^{-w^2} dw$$

by a change of variables in the integral, so that

$$P(\text{diff} > 3) = \text{erfc}(\frac{3}{2})$$

which can be looked up in tables.

8.2. Each pixel value in 500×500 pixel image $\mathcal{I}$ is an independent, normally distributed random variable with zero mean and standard deviation one. $\mathcal{I}$ is convolved with the $2k+1 \times 2k+1$ kernel $\mathcal{G}$. What is the covariance of pixel values in the result? There are two ways to do this; on a case-by-case basis (e.g. at points that are greater than $2k+1$ apart in either the x or y direction, the values are clearly independent) or in one fell swoop. Don't worry about the pixel values at the boundary.

Solution The value of each pixel in the result is a weighted sum of pixels from the input. Each pixel in the input is independent. For two pixels in the output to have non-zero covariance, they must share some elements in their sum. The covariance of two pixels with shared elements is the expected value of a product of sums, that is

$$R_{ij} = \sum_{lm} G_{lm} I_{i-l,j-m}$$

and

$$R_{uv} = \sum_{st} G_{st} I_{u-s,v-t}.$$

Now some elements of these sums are shared, and it the shared values that produce covariance. In particular, the shared terms occur when $i-l = u-s$ and $j-m = v-t$. The covariance will be the variance times the weights with which these shared terms appear. Hence

$$\mathrm{E}(R_{ij}R_{uv}) = \sum_{i-l=u-s, j-m=v-t} G_{lm}G_{st}.$$

8.3. We have a camera that can produce output values that are integers in the range from 0 to 255. Its spatial resolution is 1024 by 768 pixels, and it produces 30 frames a second. We point it at a scene that, in the absence of noise, would produce the constant value 128. The output of the camera is subject to noise that we model as zero mean stationary additive Gaussian noise with a standard deviation of 1. How long must we wait before the noise model predicts that we should see a pixel with a negative value? (Hint: You may find it helpful to use logarithms to compute the answer as a straightforward evaluation of $\exp(-128^2/2)$ will yield 0; the trick is to get the large positive and large negative logarithms to cancel.)

Solution The hint is unhelpful; DAF apologizes. Most important issue here is $P(\text{value of noise} < -128)$. This is

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-128} e^{\left(-x^2/2\right)} dx,$$

which can be looked up in tables for the complementary error function, as above. There are $30 \times 1024 \times 768$ samples per second, each of which has probability

$$P(\text{value of noise} < -128) = p$$

of having negative value. The probability of obtaining a run of samples that is N long, and contains no negative value, is $(1-p)^N$. Assume we would like a run that has a 0.9 probability of having a negative value in it; it must have at least $log(0.9)/log(1-p)$ samples in it.

8.4. We said a sensible 2D analogue to the 1D second derivative must be rotationally invariant in Section 8.3.1. Why is this true?

Solution This depends on whether we are looking for directed or undirected edges. If we look for maxima of the magnitude of the gradient, this says nothing about the direction of the edge — we have to look at the gradient magnitude for this — and so we can mark edge points by looking at local maxima without worrying about the direction of the edge. To do this with a second derivative operator, we need one that will be zero whatever the orientation of the edge; i.e. rotating the operator will not affect the response. This means it must be rotationally invariant.

Programming Assignments

8.5. Why is it necessary to check that the gradient magnitude is large at zero crossings of the Laplacian of an image? Demonstrate a series of edges for which this test is significant.

8.6. The Laplacian of a Gaussian looks similar to the difference between two Gaussians at different scales. Compare these two kernels for various values of the two scales. Which choices give a good approximation? How significant is the approximation error in edge finding using a zero-crossing approach?

8.7. Obtain an implementation of Canny's edge detector (you could try the vision home page; MATLAB has an implementation in the image processing toolbox, too) and make a series of images indicating the effects of scale and contrast thresholds on the edges that are detected. How easy is it to set up the edge detector to mark only object boundaries? Can you think of applications where this would be easy?

8.8. It is quite easy to defeat hysteresis in edge detectors that implement it — essentially, one sets the lower and higher thresholds to have the same value. Use this trick to compare the behavior of an edge detector with and without hysteresis. There are a variety of issues to look at:

(a) What are you trying to do with the edge detector output? It is sometimes helpful to have linked chains of edge points. Does hysteresis help significantly here?

(b) Noise suppression: We often wish to force edge detectors to ignore some edge points and mark others. One diagnostic that an edge is useful is high contrast (it is by no means reliable). How reliably can you use hysteresis to suppress low-contrast edges without breaking high-contrast edges?

CHAPTER 9

Texture

PROBLEMS

9.1. Show that a circle appears as an ellipse in an orthographic view, and that the minor axis of this ellipse is the tilt direction. What is the aspect ratio of this ellipse?

Solution The circle lies on a plane. An orthographic view of the plane is obtained by projecting along some family of parallel rays onto another plane. Now on the image plane there will be some direction that is parallel to the object plane — call this $\boldsymbol{T}$. Choose another direction on the image plane that is perpendicular to this one, and call it $\boldsymbol{B}$. Now I can rotate the coordinate system on the object plane without problems (it's a circle!) so I rotate it so that the x direction is parallel to $\boldsymbol{T}$. The y-coordinate projects onto the $\boldsymbol{B}$ direction (because the image plane is rotated about $\boldsymbol{T}$ with respect to the object plane) but is foreshortened. This means that the point (x, y) in the object plane projects to the point $(x, \alpha y)$ in the $\boldsymbol{T}$, $\boldsymbol{B}$ coordinate system on the image plane ($0 \leq \alpha \leq 1$ is a constant to do with the relative orientation of the planes). This means that the curve $(\cos\theta, \sin\theta)$ on the object plane goes to $(\cos\theta, \alpha\sin\theta)$ on the image plane, which is an ellipse.

9.2. We will study measuring the orientation of a plane in an orthographic view, given the texture consists of points laid down by a homogenous Poisson point process. Recall that one way to generate points according to such a process is to sample the x and y coordinate of the point uniformly and at random. We assume that the points from our process lie within a unit square.

(a) Show that the probability that a point will land in a particular set is proportional to the area of that set.

(b) Assume we partition the area into disjoint sets. Show that the number of points in each set has a multinomial probability distribution.

We will now use these observations to recover the orientation of the plane. We partition the *image texture* into a collection of disjoint sets.

(c) Show that the area of each set, *backprojected onto the textured plane*, is a function of the orientation of the plane.

(d) Use this function to suggest a method for obtaining the plane's orientation.

Solution The answer to (d) is no. The rest is straightforward.

Programming Assignments

9.3. Texture synthesis: Implement the non-parametric texture synthesis algorithm of Section 9.3.2. Use your implementation to study:

(a) the effect of window size on the synthesized texture;

(b) the effect of window shape on the synthesized texture;

(c) the effect of the matching criterion on the synthesized texture (i.e., using weighted sum of squares instead of sum of squares, etc.).

9.4. Texture representation: Implement a texture classifier that can distinguish between at least six types of texture; use the scale selection mechanism of Section 9.1.2, and compute statistics of filter outputs. We recommend that you use at least the mean and covariance of the outputs of about six oriented bar filters and a spot filter. You may need to read up on classification in chapter 22; use a simple classifier (nearest neighbor using Mahalanobis distance should do the trick).

CHAPTER 10

The Geometry of Multiple Views

PROBLEMS

10.1. Show that one of the singular values of an essential matrix is 0 and the other two are equal. (Huang and Faugeras [1989] have shown that the converse is also true—that is, any 3×3 matrix with one singular value equal to 0 and the other two equal to each other is an essential matrix.)
Hint: The singular values of $\mathcal{E}$ are the eigenvalues of $\mathcal{E}\mathcal{E}^T$.

Solution We have $\mathcal{E} = [\boldsymbol{t}_\times]\mathcal{R}$, thus $\mathcal{E}\mathcal{E}^T = [\boldsymbol{t}_\times][\boldsymbol{t}_\times]^T = [\boldsymbol{t}_\times]^T[\boldsymbol{t}_\times]$. If $\boldsymbol{a}$ is an eigenvector of $\mathcal{E}\mathcal{E}^T$ associated with the eigenvalue λ then, for any vector $\boldsymbol{b}$

$$\lambda \boldsymbol{b} \cdot \boldsymbol{a} = \boldsymbol{b}^T([\boldsymbol{t}_\times]^T[\boldsymbol{t}_\times]\boldsymbol{a}) = (\boldsymbol{t} \times \boldsymbol{b}) \cdot (\boldsymbol{t} \times \boldsymbol{a}).$$

Choosing $\boldsymbol{a} = \boldsymbol{b} = \boldsymbol{t}$ shows that $\lambda = 0$ is an eigenvalue of $\mathcal{E}\mathcal{E}^T$. Choosing $\boldsymbol{b} = \boldsymbol{t}$ shows that if $\lambda \neq 0$ then $\boldsymbol{a}$ is orthogonal to $\boldsymbol{t}$. But then choosing $\boldsymbol{a} = \boldsymbol{b}$ shows that

$$\lambda|\boldsymbol{a}|^2 = |\boldsymbol{t} \times \boldsymbol{a}|^2 = |\boldsymbol{t}|^2|\boldsymbol{a}|^2.$$

It follows that all non-zero singular values of $\mathcal{E}$ must be equal. Note that the singular values of $\mathcal{E}$ cannot all be zero since this matrix has rank 2.

10.2. Exponential representation of rotation matrices. The matrix associated with the rotation whose axis is the unit vector $\boldsymbol{a}$ and whose angle is θ can be shown to be equal to $e^{\theta[\boldsymbol{a}_\times]} \stackrel{\text{def}}{=} \sum_{i=0}^{+\infty} \frac{1}{i!}(\theta[\boldsymbol{a}_\times])^i$. Use this representation to derive Eq. (10.3).

Solution Let us consider a small motion with translational velocity $\boldsymbol{v}$ and rotational velocity $\boldsymbol{\omega}$. If the two camera frames are separated by the small time interval δt, the translation separating them is obviously (to first order) $\boldsymbol{t} = \delta t \boldsymbol{v}$. The corresponding rotation is a rotation of angle $\delta t|\boldsymbol{\omega}|$ about the axis $(1/|\boldsymbol{\omega}|)\boldsymbol{\omega}$, i.e.,

$$\mathcal{R} = e^{\delta t[\boldsymbol{\omega}_\times]} = \sum_{i=0}^{+\infty} \frac{1}{i!}(\delta t[\boldsymbol{\omega}_\times])^i = \text{Id} + \delta t\,[\boldsymbol{\omega}_\times] + \text{higher-order terms.}$$

Neglecting all terms of order two or higher yields Eq. (10.3).

10.3. The infinitesimal epipolar constraint of Eq. (10.4) was derived by assuming that the observed scene was static and the camera was moving. Show that when the camera is fixed and the scene is moving with translational velocity $\boldsymbol{v}$ and rotational velocity $\boldsymbol{\omega}$, the epipolar constraint can be rewritten as $\boldsymbol{p}^T([\boldsymbol{v}_\times][\boldsymbol{\omega}_\times])\boldsymbol{p} + (\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{v} = 0$. Note that this equation is now the sum of the two terms appearing in Eq. (10.4) instead of their difference.
Hint: If $\mathcal{R}$ and $\boldsymbol{t}$ denote the rotation matrix and translation vectors appearing in the definition of the essential matrix for a moving camera, show that the object displacement that yields the same motion field for a static camera is given by the rotation matrix $\mathcal{R}^T$ and the translation vector $-\mathcal{R}^T\boldsymbol{t}$.

Solution Let us consider first a moving camera and a static scene, use the coordinate system attached to the camera in its initial position as the world coordinate system, and identify scene points with their positions in this coordinate system and image points with their position in the corresponding camera coordinate system.
We have seen that the projection matrix associated with this camera can be taken equal to $\mathcal{M} = \begin{pmatrix}\mathrm{Id} & \mathbf{0}\end{pmatrix}$ before the motion and to $\mathcal{M} = \begin{pmatrix}\mathcal{R}_c^T & -\mathcal{R}_c^T \boldsymbol{t}_c\end{pmatrix}$ after the camera has undergone a rotation $\mathcal{R}_c$ and a translation $\boldsymbol{t}_c$. Using non-homogeneous coordinates for scene points and homogeneous ones for image points, the two images of a point $\boldsymbol{P}$ are thus $\boldsymbol{p} = \boldsymbol{P}$ and $\boldsymbol{p}' = \mathcal{R}_c^T \boldsymbol{P} - \mathcal{R}_c^T \boldsymbol{t}_c$.
Let us now consider a static camera and a moving object. Suppose this object undergoes the (finite) motion defined by $P' = \mathcal{R}_o \boldsymbol{P} + \boldsymbol{t}_o$ in the coordinate system attached to this camera. Since the projection matrix is $\begin{pmatrix}\mathrm{Id} & \mathbf{0}\end{pmatrix}$ in this coordinate system, the image of $\boldsymbol{P}$ before the object displacement is $\boldsymbol{p} = \boldsymbol{P}$. The image after the displacement is $\boldsymbol{p}' = \mathcal{R}_o \boldsymbol{P} + \boldsymbol{t}_o$, and it follows immediately that taking $\mathcal{R}_o = \mathcal{R}^T$ and $\boldsymbol{t}_o = -\mathcal{R}^T \boldsymbol{t}_c$ yields the same motion field as before.
For small motions, we have

$$\mathcal{R}_o = \mathrm{Id} + \delta t\, [\boldsymbol{\omega}_{o\times}] = \mathcal{R}_c^T = \mathrm{Id} - \delta t\, [\boldsymbol{\omega}_{c\times}],$$

and it follows that $\boldsymbol{\omega}_o = -\boldsymbol{\omega}_c$. Likewise,

$$\boldsymbol{t}_o = \delta t\, \boldsymbol{v}_o = -\mathcal{R}_c^T \boldsymbol{t}_c = -(\mathrm{Id} - \delta t\, [\boldsymbol{\omega}_{c\times}])(\delta t\, \boldsymbol{v}_c) = -\delta t\, \boldsymbol{v}_c$$

when second-order terms are neglected. Thus $\boldsymbol{v}_c = -\boldsymbol{v}_o$. Recall that Eq. (10.4) can be written as

$$\boldsymbol{p}^T([\boldsymbol{v}_{c\times}][\boldsymbol{\omega}_{c\times}])\boldsymbol{p} - (\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{v}_c = 0.$$

Substituting $\boldsymbol{v}_c = -\boldsymbol{v}_o$ and $\boldsymbol{\omega}_c = -\boldsymbol{\omega}_o$ in this equation finally yields

$$\boldsymbol{p}^T([\boldsymbol{v}_{o\times}][\boldsymbol{\omega}_{o\times}])\boldsymbol{p} + (\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{v}_o = 0.$$

10.4. Show that when the 8×8 matrix associated with the eight-point algorithm is singular, the eight points and the two optical centers lie on a quadric surface (Faugeras, 1993).
Hint: Use the fact that when a matrix is singular, there exists some nontrivial linear combination of its columns that is equal to zero. Also take advantage of the fact that the matrices representing the two projections in the coordinate system of the first camera are in this case $(\mathrm{Id} \quad \mathbf{0})$ and $(\mathcal{R}^T \quad -\mathcal{R}^T \boldsymbol{t})$.

Solution We follow the proof in Faugeras (1993): Each row of the 8×8 matrix associated with the eight-point algorithm can be written as

$$(uu', uv', u, vu', vv', v, u', v') = \frac{1}{zz'}(xx', xy', xz', yx', yy', yz', zx', zy'),$$

where $\boldsymbol{P} = (x, y, z)^T$ and $\boldsymbol{P}' = (x', y', z')^T$ denote the positions of the scene point projecting onto $(u, v)^T$ and $(u', v')^T$ in the corresponding camera coordinate systems (C) and (C'). For the matrix to be singular, there must exist some nontrivial linear combination of its columns that is equal to zero—that is, there must exist eight scalars λ_i $(i = 1, \ldots, 8)$ such that

$$\lambda_1 xx' + \lambda_2 xy' + \lambda_3 xz' + \lambda_4 yx' + \lambda_5 yy' + \lambda_6 yz' + \lambda_7 zx' + \lambda_8 zy' = 0,$$

or, in matrix form,

$$\boldsymbol{P}^T \mathcal{Q} \boldsymbol{P}' = 0, \quad \text{where} \quad \mathcal{Q} = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_4 & \lambda_5 & \lambda_6 \\ \lambda_7 & \lambda_8 & 0 \end{pmatrix}.$$

Now, with the conventions used to define $\mathcal{R}$ and $\boldsymbol{t}$, we have $\mathcal{R} = {}^{C}_{C'}\mathcal{R}$ and $\boldsymbol{t} = {}^{C}O' = {}^{C}\overrightarrow{OO'}$, where O and O' denote the optical centers of the two cameras. It follows that

$$\boldsymbol{P}' = {}^{C'}P = {}^{C'}_{C}\mathcal{R}^{C}P + {}^{C'}\overrightarrow{O'O} = {}^{C'}_{C}\mathcal{R}^{C}P - {}^{C'}\overrightarrow{OO'} = {}^{C'}_{C}\mathcal{R}^{C}P - {}^{C'}_{C}\mathcal{R}^{C}\overrightarrow{OO'} = \mathcal{R}^T(\boldsymbol{P} - \boldsymbol{t}).$$

The degeneracy condition derived earlier can therefore be written as

$$\boldsymbol{P}^T \mathcal{Q} \mathcal{R}^T (\boldsymbol{P} - \boldsymbol{t}) = 0.$$

This equation is quadratic in the coordinates of $\boldsymbol{P}$ and defines a quadric surface in the coordinate system attached to the first camera. This quadric passes through the two optical centers since its equation is obviously satisfied by $\boldsymbol{P} = \boldsymbol{0}$ and $\boldsymbol{P} = \boldsymbol{t}$. When the 8×8 matrix involved in the eight-point algorithm is singular, the eight points must therefore lie on a quadric passing through these two points.

10.5. Show that three of the determinants of the 3×3 minors of

$$\mathcal{L} = \begin{pmatrix} \boldsymbol{l}_1^T & 0 \\ \boldsymbol{l}_2^T \mathcal{R}_2 & \boldsymbol{l}_2^T \boldsymbol{t}_2 \\ \boldsymbol{l}_3^T \mathcal{R}_3 & \boldsymbol{l}_3^T \boldsymbol{t}_3 \end{pmatrix} \quad \text{can be written as} \quad \boldsymbol{l}_1 \times \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix} = \boldsymbol{0}.$$

Show that the fourth determinant can be written as a linear combination of these.

Solution Let us first note that

$$\begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix} = \begin{pmatrix} (\boldsymbol{l}_2^T \boldsymbol{t}_2)(\boldsymbol{R}_3^{1T} \boldsymbol{l}_3) - (\boldsymbol{l}_2^T \boldsymbol{R}_2^1)(\boldsymbol{t}_3^T \boldsymbol{l}_3) \\ (\boldsymbol{l}_2^T \boldsymbol{t}_2)(\boldsymbol{R}_3^{2T} \boldsymbol{l}_3) - (\boldsymbol{l}_2^T \boldsymbol{R}_2^2)(\boldsymbol{t}_3^T \boldsymbol{l}_3) \\ (\boldsymbol{l}_2^T \boldsymbol{t}_2)(\boldsymbol{R}_3^{3T} \boldsymbol{l}_3) - (\boldsymbol{l}_2^T \boldsymbol{R}_2^3)(\boldsymbol{t}_3^T \boldsymbol{l}_3) \end{pmatrix} = (\boldsymbol{l}_2^T \boldsymbol{t}_2)(\mathcal{R}_3^T \boldsymbol{l}_3) - (\boldsymbol{l}_3^T \boldsymbol{t}_3)(\mathcal{R}_2^T \boldsymbol{l}_2).$$

To simplify notation, we now introduce the three vectors $\boldsymbol{a} = \boldsymbol{l}_1$, $\boldsymbol{b} = \mathcal{R}_2^T \boldsymbol{l}_2$, and $\boldsymbol{c} = \mathcal{R}_3^T \boldsymbol{l}_3$, and the two scalars $d = \boldsymbol{t}_2^T \boldsymbol{l}_2$ and $e = \boldsymbol{t}_3^T \boldsymbol{l}_3$. With this notation, we can now rewrite the trifocal constraints as

$$\boldsymbol{a} \times [d\boldsymbol{c} - e\boldsymbol{b}] = d(\boldsymbol{a} \times \boldsymbol{c}) - e(\boldsymbol{a} \times \boldsymbol{b}) = 0.$$

With the same notation, we have

$$\mathcal{L}^T = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} \\ 0 & d & e \end{pmatrix}$$

Let us now compute the determinants of the 3×3 minors of $\mathcal{L}^T$ (instead of those of $\mathcal{L}$; this is just to avoid transposing too many vectors and does not change the result of our derivstion).

Three of these determinants can be written as

$$D_{12} = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & d & e \end{vmatrix}, \ D_{23} = \begin{vmatrix} a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ 0 & d & e \end{vmatrix}, \ D_{31} = \begin{vmatrix} a_3 & b_3 & c_3 \\ a_1 & b_1 & c_1 \\ 0 & d & e \end{vmatrix}.$$

In particular,

$$\begin{pmatrix} D_{23} \\ D_{31} \\ D_{12} \end{pmatrix} = \begin{pmatrix} (a_2b_3 - a_3b_2)e - (a_2c_3 - a_3c_2)d \\ (a_3b_1 - a_1b_3)e - (a_3c_1 - a_1c_3)d \\ (a_1b_2 - a_2b_1)e - (a_1c_2 - a_2c_1)d \end{pmatrix} = e(\boldsymbol{a} \times \boldsymbol{b}) - d(\boldsymbol{a} \times \boldsymbol{c}),$$

which, except for a change in sign, is identical to the expression derived earlier. Thus the fact that three of the 3×3 minors of $\mathcal{L}$ are zero can indeed be expressed by the trifocal tensor.

Let us conclude by showing that the fourth determinant is a linear combination of the other three. This determinant is

$$D = \begin{vmatrix} \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} \end{vmatrix} = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c}.$$

But $(\boldsymbol{a} \times \boldsymbol{c}) \cdot \boldsymbol{c} = 0$, thus we can write

$$D = \frac{1}{e}[e(\boldsymbol{a} \times \boldsymbol{b}) - d(\boldsymbol{a} \times \boldsymbol{c})] \cdot \boldsymbol{c} = \frac{c_1}{e} D_{23} + \frac{c_2}{e} D_{31} + \frac{c_3}{e} D_{12},$$

which shows that D can indeed be written as a linear combination of D_{12}, D_{23}, and D_{31}.

10.6. Show that Eq. (10.18) reduces to Eq. (10.2) when $\mathcal{M}_1 = (\mathrm{Id} \quad \mathbf{0})$ and $\mathcal{M}_2 = (\mathcal{R}^T \quad -\mathcal{R}^T\boldsymbol{t})$.

Solution Recall that Eq. (10.18) expresses the bilinear constraints associated with two cameras as $D = 0$, where D is the determinant

$$D = \begin{vmatrix} u_1\mathcal{M}_1^3 - \mathcal{M}_1^1 \\ v_1\mathcal{M}_1^3 - \mathcal{M}_1^2 \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_2\mathcal{M}_2^3 - \mathcal{M}_2^2 \end{vmatrix},$$

and $\mathcal{M}_i^j$ denotes row number j of camera number i. When

$$\mathcal{M}_1 = \begin{pmatrix} \mathrm{Id} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \text{ and } \mathcal{M}_2 = \begin{pmatrix} \mathcal{R}^T & -\mathcal{R}^T\boldsymbol{t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_1^T & -\boldsymbol{c}_1 \cdot \boldsymbol{t} \\ \boldsymbol{c}_2^T & -\boldsymbol{c}_2 \cdot \boldsymbol{t} \\ \boldsymbol{c}_3^T & -\boldsymbol{c}_3 \cdot \boldsymbol{t} \end{pmatrix},$$

where $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, and $\boldsymbol{c}_3$ denote the three *columns* of $\mathcal{R}$, we can rewrite the determinant as

$$\begin{aligned}
D &= \begin{vmatrix} (-1, 0, u_1) & 0 \\ (0, -1, v_1) & 0 \\ u_2\boldsymbol{c}_3^T - \boldsymbol{c}_1^T & -(u_2\boldsymbol{c}_3 - \boldsymbol{c}_1) \cdot \boldsymbol{t} \\ v_2\boldsymbol{c}_3^T - \boldsymbol{c}_2^T & -(v_2\boldsymbol{c}_3 - \boldsymbol{c}_2) \cdot \boldsymbol{t} \end{vmatrix} \\
&= \begin{vmatrix} -1 & 0 & \\ 0 & -1 & v_2\boldsymbol{c}_3 - \boldsymbol{c}_2 \\ u_1 & v_1 & \end{vmatrix} (u_2\boldsymbol{c}_3 - \boldsymbol{c}_1) \cdot \boldsymbol{t} - \begin{vmatrix} -1 & 0 & \\ 0 & -1 & u_2\boldsymbol{c}_3 - \boldsymbol{c}_1 \\ u_1 & v_1 & \end{vmatrix} (v_2\boldsymbol{c}_3 - \boldsymbol{c}_2) \cdot \boldsymbol{t} \\
&= [\boldsymbol{p}_1 \cdot (v_2\boldsymbol{c}_3 - \boldsymbol{c}_2)][(u_2\boldsymbol{c}_3 - \boldsymbol{c}_1) \cdot \boldsymbol{t}] - [\boldsymbol{p}_1 \cdot (u_2\boldsymbol{c}_3 - \boldsymbol{c}_1)][(v_2\boldsymbol{c}_3 - \boldsymbol{c}_2) \cdot \boldsymbol{t}] \\
&= \boldsymbol{p}_1^T \big((\boldsymbol{c}_2 \cdot \boldsymbol{t})\boldsymbol{c}_3 - (\boldsymbol{c}_3 \cdot \boldsymbol{t})\boldsymbol{c}_2 \quad (\boldsymbol{c}_3 \cdot \boldsymbol{t})\boldsymbol{c}_1 - (\boldsymbol{c}_1 \cdot \boldsymbol{t})\boldsymbol{c}_3 \quad (\boldsymbol{c}_1 \cdot \boldsymbol{t})\boldsymbol{c}_2 - (\boldsymbol{c}_2 \cdot \boldsymbol{t})\boldsymbol{c}_1 \big) \boldsymbol{p}_2,
\end{aligned}$$

where $\boldsymbol{p}_1 = (u_1, v_1, 1)^T$ and $\boldsymbol{p}_2 = (u_2, v_2, 1)^T$.
Now, let (t_1, t_2, t_3) be the coordinates of the vector $\boldsymbol{t}$ in the right-handed orthonormal basis formed by the columns of $\mathcal{R}$, we have

$$\begin{aligned} D &= \boldsymbol{p}_1^T \big(t_2\boldsymbol{c}_3 - t_3\boldsymbol{c}_2 \quad t_3\boldsymbol{c}_1 - t_1\boldsymbol{c}_3 \quad t_1\boldsymbol{c}_2 - t_2\boldsymbol{c}_1 \big) \boldsymbol{p}_2 \\ &= \boldsymbol{p}_1^T \big(\boldsymbol{t} \times \boldsymbol{c}_1 \quad \boldsymbol{t} \times \boldsymbol{c}_2 \quad \boldsymbol{t} \times \boldsymbol{c}_3 \big) \boldsymbol{p}_2 = \boldsymbol{p}_1^T [\boldsymbol{t}_\times] \mathcal{R} \boldsymbol{p}_2, \end{aligned}$$

which is indeed an instance of Eq. (10.2).

10.7. Show that Eq. (10.19) reduces to Eq. (10.15) when $\mathcal{M}_1 = (\mathrm{Id} \quad \mathbf{0})$.

Solution Recall that Eq. (10.18) expresses the trilinear constraints associated with two cameras as $D = 0$, where D is the determinant

$$D = \begin{vmatrix} u_1\mathcal{M}_1^3 - \mathcal{M}_1^1 \\ v_1\mathcal{M}_1^3 - \mathcal{M}_1^2 \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \end{vmatrix},$$

and $\mathcal{M}_i^j$ denotes row number j of camera number i. When $\mathcal{M}_1 = \big(\mathrm{Id} \quad \mathbf{0}\big)$,

$$D = \begin{vmatrix} (-1, 0, u_1, 0) \\ (0, -1, v_1, 0) \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \end{vmatrix}.$$

Introducing $\boldsymbol{l}_1 = (0, -1, v_1)^T$, $\boldsymbol{l}_2 = (-1, 0, u_2)^T$, and $\boldsymbol{l}_3 = (0, -1, v_3)^T$ allows us to rewrite this equation as

$$D = \begin{vmatrix} (-1, 0, u_1, 0) \\ \boldsymbol{l}_1{}^T \mathcal{M}_1 \\ \boldsymbol{l}_2{}^T \mathcal{M}_2 \\ \boldsymbol{l}_3{}^T \mathcal{M}_3 \end{vmatrix},$$

or, since the determinant of a matrix is equal to the determinant of its transpose,

$$D = \begin{vmatrix} \begin{pmatrix} -1 \\ 0 \\ u_1 \end{pmatrix} & \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} \\ 0 & 0 & d & e \end{vmatrix} = (-1, 0, u_1) \begin{pmatrix} D_{23} \\ D_{31} \\ D_{12} \end{pmatrix},$$

where we use the same notation as in Ex. 10.5. According to that exercise, we thus have

$$\begin{aligned} D &= -(-1, 0, u_1)[\boldsymbol{l}_1 \times \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix}] = -(-1, 0, u_1)[\boldsymbol{l}_1 \times] \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix} \\ &= -(\begin{pmatrix} -1 \\ 0 \\ u_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ -1 \\ v_1 \end{pmatrix}) \cdot \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix} = -\boldsymbol{p}_1^T \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix}, \end{aligned}$$

which is the result we aimed to prove.

10.8. Develop Eq. (10.20) with respect to the image coordinates, and verify that the coefficients can indeed be written in the form of Eq. (10.21).

Solution This follows directly from the multilinear nature of determinants. Indeed, Eq. (10.20) can be written as

$$0 = \begin{vmatrix} v_1\mathcal{M}_1^3 - \mathcal{M}_1^2 \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix} = v_1 \begin{vmatrix} \mathcal{M}_1^3 \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix} - \begin{vmatrix} \mathcal{M}_1^2 \\ u_2\mathcal{M}_2^3 - \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix}$$

$$= v_1 u_2 \begin{vmatrix} \mathcal{M}_1^3 \\ \mathcal{M}_2^3 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix} - v_1 \begin{vmatrix} \mathcal{M}_1^3 \\ \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix} - u_2 \begin{vmatrix} \mathcal{M}_1^2 \\ \mathcal{M}_2^3 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix} + \begin{vmatrix} \mathcal{M}_1^2 \\ \mathcal{M}_2^1 \\ v_3\mathcal{M}_3^3 - \mathcal{M}_3^2 \\ v_4\mathcal{M}_4^3 - \mathcal{M}_4^2 \end{vmatrix}, \text{etc.}$$

It is thus clear that all the coefficients of the quadrilinear tensor can be written in the form of Eq. (10.21).

10.9. Use Eq. (10.23) to calculate the unknowns z_i, λ_i, and z_1^i in terms of $\boldsymbol{p}_1$, $\boldsymbol{p}_i$, $\mathcal{R}_i$, and $\boldsymbol{t}_i$ ($i = 2, 3$). Show that the value of λ_i is directly related to the epipolar constraint, and characterize the degree of the dependency of $z_1^2 - z_1^3$ on the data points.

Solution We rewrite Eq. (10.23) as

$$z_1^i \boldsymbol{p}_1 = \boldsymbol{t}_i + z_i \mathcal{R}_i \boldsymbol{p}_i + \lambda_i (\boldsymbol{p}_1 \times \mathcal{R}_i \boldsymbol{p}_i) \Longleftrightarrow z_1^i \boldsymbol{p}_1 = \boldsymbol{t}_i + z_i \boldsymbol{q}_i + \lambda_i \boldsymbol{r}_i,$$

where $\boldsymbol{q}_i = \mathcal{R}_i \boldsymbol{p}_i$, $\boldsymbol{r}_i = \boldsymbol{p}_1 \times \boldsymbol{q}_i$, and $i = 2, 3$.
Forming the dot product of this equation with $\boldsymbol{r}_i$ yields

$$\boldsymbol{t}_i \cdot [\boldsymbol{p}_1 \times \mathcal{R}_i \boldsymbol{p}_i] + \lambda_i |\boldsymbol{r}_i|^2 = 0,$$

or, rearranging the terms in the triple product,

$$\lambda_i |\boldsymbol{r}_i|^2 = \boldsymbol{p}_1 \cdot [\boldsymbol{t}_i \times \mathcal{R}_i \boldsymbol{p}_i],$$

which can be rewritten as

$$\lambda_i = \frac{\boldsymbol{p}_1^T \mathcal{E}_i \boldsymbol{p}_i}{|\boldsymbol{r}_i|^2},$$

where $\mathcal{E}_i$ is the essential matrix associated with views number 1 and i. So the value of λ_i is indeed directly related to the epipolar constraint. In particular, if this constraint is exactly satisfied, $\lambda_i = 0$.
Now forming the dot product of Eq. (10.23) with $\boldsymbol{q}_i \times \boldsymbol{r}_i$ yields

$$z_1^i \boldsymbol{p}_1 \cdot (\boldsymbol{q}_i \times \boldsymbol{r}_i) = \boldsymbol{t}_i \cdot (\boldsymbol{q}_i \times \boldsymbol{r}_i).$$

Let us denote by $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}] = \boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c})$ the triple product of vectors in $\mathbb{R}^3$. Noting that the value of the triple product is invariant under circular permutations of the vectors reduces the above equation to

$$z_1^i \boldsymbol{r}_i \cdot (\boldsymbol{p}_1 \times \boldsymbol{q}_i) = \boldsymbol{r}_i \cdot (\boldsymbol{t}_i \times \boldsymbol{q}_i),$$

or

$$z_1^i = \frac{\boldsymbol{r}_i^T \mathcal{E}_i \boldsymbol{p}_i}{|\boldsymbol{r}_i|^2}.$$

Likewise, forming the dot product of Eq. (10.23) with $\boldsymbol{p}_1 \times \boldsymbol{r}_i$ yields

$$[\boldsymbol{t}_i, \boldsymbol{p}_1, \boldsymbol{r}_i] + z_i[\boldsymbol{q}_i, \boldsymbol{p}_1, \boldsymbol{r}_1] = 0,$$

or

$$z_i = \frac{[\boldsymbol{t}_i, \boldsymbol{p}_1, \boldsymbol{r}_i]}{|\boldsymbol{r}_i|^2}.$$

The scale-restraint condition can be written as $z_1^2 - z_1^3 = 0$, or

$$|\boldsymbol{r}_2|^2(\boldsymbol{r}_3^T \mathcal{E}_3 \boldsymbol{p}_3) = |\boldsymbol{r}_3|^2(\boldsymbol{r}_2^T \mathcal{E}_2 \boldsymbol{p}_2),$$

which is an algebraic condition of degree 7 in $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$.

Programming Assignments

10.10. Implement the eight-point algorithm for weak calibration from binocular point correspondences.

10.11. Implement the linear least-squares version of that algorithm with and without Hartley's preconditioning step.

10.12. Implement an algorithm for estimating the trifocal tensor from point correspondences.

10.13. Implement an algorithm for estimating the trifocal tensor from line correspondences.

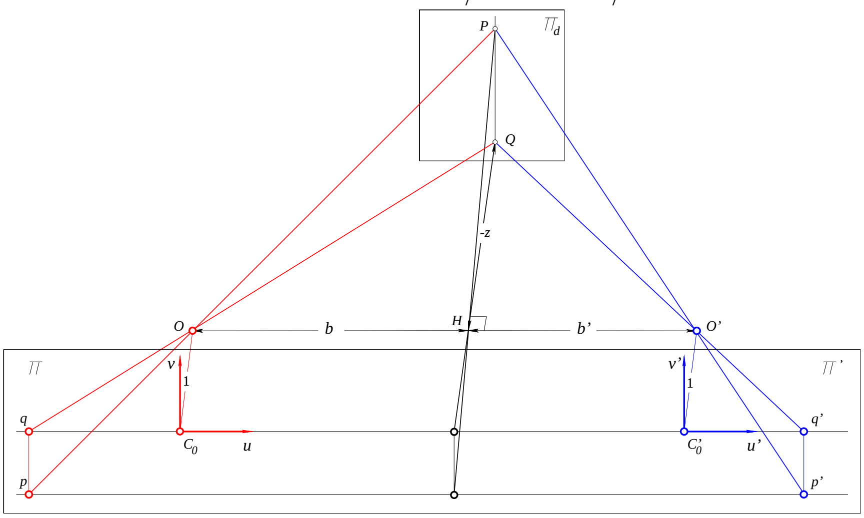
CHAPTER 11

Stereopsis

PROBLEMS

11.1. Show that, in the case of a rectified pair of images, the depth of a point P in the normalized coordinate system attached to the first camera is $z = -B/d$, where B is the baseline and d is the disparity.

Solution Note that for rectified cameras, the v and v' axis of the two image coordinate systems are parallel to each other and to the y axis of the coordinate system attached to the first camera. In addition, the images q and q' of any point Q in the plane $y = 0$ verify $v = v' = 0$. As shown by the diagram below, if H, C_0 and C_0' denote respectively the orthogonal projection of Q onto the baseline and the principal points of the two cameras, the triangles OHQ and qC_0O are similar, thus $-b/z = -u/1$. Likewise, the triangles $HO'Q$ and $C_0'q'O'$ are similar, thus $-b'/z = u'/1$, where b and b' denote respectively the lengths of the line segments OH and HO'. It follows that $u' - u = -B/z$ or $z = -B/d$.



Let us now consider a point P with nonzero y coordinate and its orthogonal projection Q onto the plane $y = 0$. The points P and Q have the same depth since the line PQ joining them is parallel to the y axis. The lines pq and $p'q'$ joining the projections of the two points in the two images are also obviously parallel to PQ and to the v and v' axis. It follows that the u coordinates of p and q are the same, and that the u' coordinates of p' and q' are also the same. In other words, the disparity and depths for the points P and Q are the same, and the formula $z = -B/d$ holds in general.

11.2. Use the definition of disparity to characterize the accuracy of stereo reconstruction as a function of baseline and depth.

Solution Let us assume that the cameras have been rectified. In this case, as in

Ex. 11.1, we have $z = -B/d$. Let us assume the disparity has been measured with some error ε. Using a first-order Taylor expansion of the depth shows that

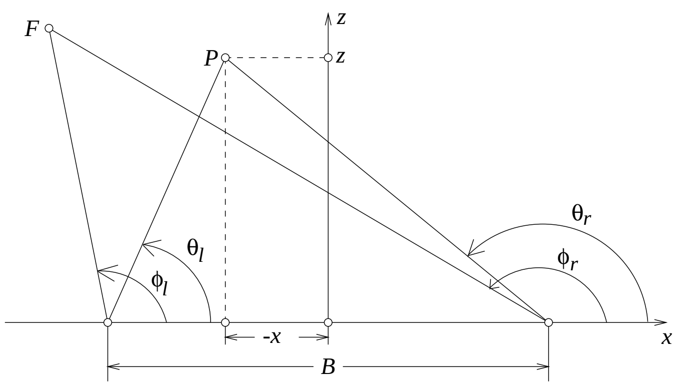
$$z(\delta+\varepsilon) - z(\delta) \approx \varepsilon z'(\delta) = \varepsilon \frac{B}{d^2} = \frac{\varepsilon}{B} z^2.$$

In other words, the error is proportional to the squared depth and inversely proportional to the baseline.

11.3. Give reconstruction formulas for verging eyes in the plane.

Solution Let us define a Cyclopean coordinate system with origin at the midpoint between the two eyes, x axis in the direction of the baseline, and z axis oriented so that $z > 0$ for points in front of the two eyes (note that this contradicts our usual conventions, but allows us to use a right-handed (x, z) coordinate system). Now consider a point P with coordinates (x, z). As shown by the diagram below, if the corresponding projection rays make angles θ_l and θ_r with the z axis, we must have

$$\begin{cases} \dfrac{x + B/2}{z} = \cot\theta_l, \\ \dfrac{x - B/2}{z} = \cot\theta_r, \end{cases} \iff \begin{cases} x = \dfrac{B}{2}\dfrac{\cot\theta_l + \cot\theta_r}{\cot\theta_l - \cot\theta_r}, \\ z = \dfrac{B}{\cot\theta_l - \cot\theta_r}. \end{cases}$$



Now, if the fixated point has angular coordinates (ϕ_l, ϕ_r) and some other point P has absolute angular coordinates (θ_l, θ_r), Cartesian coordinates (x, z), and retinal angular coordinates (ψ_l, ψ_r), we must have $\theta_l = \phi_l + \psi_l$ and $\theta_r = \phi_r + \psi_r$, which gives reconstruction formulas for given values of (ϕ_l, ϕ_r) and (ψ_l, ψ_r).

11.4. Give an algorithm for generating an ambiguous random dot stereogram that can depict two different planes hovering over a third one.

Solution We display two squares hovering at different heights over a larger bacground square. The background images can be synthesized by spraying random black dots on a white background plate after (virtually) covering the area corresponding to the hovering squares. For the other two squares, the dots are generated as follows: On a given scanline, intersect a ray issued from the left eye with the first plane and the second one, and paint black the resulting dots P_1 and P_2. Then paint a black dot on the first plane at the point P_3 where the ray joining the right eye to P_2 intersects the first plane. Now intersect the ray joining the left eye to P_3 with the second plane. Continue this process as long as desired. It is clear that this will generate a deterministic, but completely ambiguous pattern. Limiting this process to a few iterations and repeating it at many random locations will achieve the desired random effect.

11.5. Show that the correlation function reaches its maximum value of 1 when the image brightnesses of the two windows are related by the affine transform $I' = \lambda I + \mu$ for some constants λ and μ with $\lambda > 0$.

Solution Let us consider two images represented by the vectors $\boldsymbol{w} = (w_1, \ldots, w_p)^T$ and $\boldsymbol{w}' = (w'_1, \ldots, w'_p)^T$ of $\mathbb{R}^p$ (typically, $p = (2m+1) \times (2n+1)$ for some positive values of m and n). As noted earlier, the corresponding normalized correlation value is the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$, where $\bar{\boldsymbol{a}}$ denotes the vector whose coordinates are all equal to the mean $\bar{a}$ of the coordinates of $\boldsymbol{a}$.

The correlation function reaches its maximum value of 1 when the angle θ is zero. In this case, we must have $\boldsymbol{w}' - \bar{\boldsymbol{w}}' = \lambda(\boldsymbol{w} - \bar{\boldsymbol{w}})$ for some $\lambda > 0$, or for $i = 1, \ldots, p$,

$$w'_i = \lambda w_i + (\bar{w}' - \lambda \bar{w}) = \lambda w_i + \mu,$$

where $\mu = \bar{w}' - \lambda \bar{w}$.

Conversely, suppose that $w'_i = \lambda w_i + \mu$ for some λ, μ with $\lambda > 0$. Clearly, $\boldsymbol{w}' = \lambda \boldsymbol{w} + \bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{w}}' = \lambda \bar{\boldsymbol{w}} + \bar{\boldsymbol{\mu}}$, where $\bar{\boldsymbol{\mu}}$ denotes this time the vector with all coordinates equal to μ. Thus $\boldsymbol{w}' - \bar{\boldsymbol{w}}' = \lambda(\boldsymbol{w} - \bar{\boldsymbol{w}})$, and the angle θ is equal to zero, yielding the maximum possible value of the correlation function.

11.6. Prove the equivalence of correlation and sum of squared differences for images with zero mean and unit Frobenius norm.

Solution Let $\boldsymbol{w}$ and $\boldsymbol{w}'$ denote the vectors associated with two image windows. If these windows have zero mean and unit Frobenius norm, we have by definition $|\boldsymbol{w}|^2 = |\boldsymbol{w}'|^2 = 1$ and $\bar{\boldsymbol{w}} = \bar{\boldsymbol{w}}' = 0$. In this case, the sum of squared differences is

$$|\boldsymbol{w}' - \boldsymbol{w}|^2 = |\boldsymbol{w}|^2 - 2\boldsymbol{w} \cdot \boldsymbol{w}' + |\boldsymbol{w}'|^2 = 2 - 2\boldsymbol{w} \cdot \boldsymbol{w}' = 2 - 2C,$$

where C is the normalized correlation of the two windows. Thus minimizing the sum of squared differences is equivalent to maximizing the normalized correlation.

11.7. Recursive computation of the correlation function.

(a) Show that $(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}') = \boldsymbol{w} \cdot \boldsymbol{w}' - (2m+1)(2n+1)\bar{I}\bar{I}'$.

(b) Show that the average intensity $\bar{I}$ can be computed recursively, and estimate the cost of the incremental computation.

(c) Generalize the prior calculations to all elements involved in the construction of the correlation function, and estimate the overall cost of correlation over a pair of images.

Solution

(a) First note that for any two vectors of size p, we have $\bar{\boldsymbol{a}} \cdot \boldsymbol{b} = p\bar{a}\bar{b}$, where $\bar{a}$ and $\bar{b}$ denote respectively the average values of the coordinates of $\boldsymbol{a}$ and $\boldsymbol{b}$. For vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ representing images of size $(2m+1) \times (2n+1)$ with average intensities $\bar{I}$ and $\bar{I}'$, we have therefore

$$\begin{aligned}(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}') &= \boldsymbol{w} \cdot \boldsymbol{w}' - \bar{\boldsymbol{w}} \cdot \boldsymbol{w}' - \boldsymbol{w} \cdot \bar{\boldsymbol{w}}' + \bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{w}}' \\ &= \boldsymbol{w} \cdot \boldsymbol{w}' - (2m+1)(2n+1)\bar{I}\bar{I}'.\end{aligned}$$

(b) Let $\bar{I}(i,j)$ and $\bar{I}(i+1,j)$ denote the average intensities computed for windows respectively centered in (i,j) and $(i+1,j)$. If $p = (2m+1) \times (2n+1)$, we have

$$
\begin{aligned}
\bar{I}(i+1,j) &= \frac{1}{p}\sum_{k=-m}^{m}\sum_{l=-n}^{n} I(i+k+1,j+l) \\
&= \frac{1}{p}\sum_{k=-m}^{m-1}\sum_{l=-n}^{n} I(i+k+1,j+l) + \frac{1}{p}\sum_{l=-n}^{n} I(i+m+1,j+l) \\
&= \frac{1}{p}\sum_{k'=-m+1}^{m}\sum_{l=-n}^{n} I(i+k',j+l) + \frac{1}{p}\sum_{l=-n}^{n} I(i+m+1,j+l) \\
&= \bar{I}(i,j) - \frac{1}{p}\sum_{l=-n}^{n} I(i-m,j+l) + \frac{1}{p}\sum_{l=-n}^{n} I(i+m+1,j+l).
\end{aligned}
$$

Thus the average intensity can be updated in $4(n+1)$ operations when moving from one pixel to the one below it. The update for moving one column to the right costs $4(m+1)$ operations. This is to compare to the $(2m+1)(2n+1)$ operations necessary to compute the average from scratch.

(c) It is not possible to compute the dot product incrementally during column shifts associated with successive disparities. However, it is possible to compute the dot product associated with elementary row shifts since $2m$ of the rows are shared by consecutive windows. Indeed, let $\boldsymbol{w}(i,j)$ and $\boldsymbol{w}'(i,j)$ denotes the vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ associated with windows of size $(2m+1)\times(2n+1)$ centered in (i,j). We have

$$
\boldsymbol{w}(i+1,j)\cdot\boldsymbol{w}'(i+1,j) = \sum_{k=-m}^{m}\sum_{l=-n}^{n} I(i+k+1,j+l)I'(i+k+1,j+l),
$$

and the exact same line of reasoning as in (b) can be used to show that

$$
\begin{aligned}
\boldsymbol{w}(i+1,j)\cdot\boldsymbol{w}'(i+1,j) &= \boldsymbol{w}(i,j)\cdot\boldsymbol{w}'(i,j) \\
&- \sum_{l=-n}^{n} I(i-m,j+l)I'(i-m,j+l) \\
&+ \sum_{l=-n}^{n} I(i+m+1,j+l)I'(i+m+1,j+l).
\end{aligned}
$$

Thus the dot product can be updated in $4(2n+1)$ operations when moving from one pixel to the one below it. This is to compare to the $2(2m+1)(2n+1)-1$ operations necessary to compute the dot product from scratch.

To complete the computation of the correlation function, one must also compute the norms $|\boldsymbol{w}-\bar{\boldsymbol{w}}|$ and $|\boldsymbol{w}'-\bar{\boldsymbol{w}}'|$. This computation also reduces to the evaluation of a dot product and an average, but it can be done recursively for both rows and column shifts.

Suppose that images are matched by searching for each pixel in the left image its match in the same scanline of the right image, within some disparity range $[-D,D]$. Suppose also that the two images have size $M\times N$ and that the windows being compared have, as before, size $(2m+1)\times(2n+1)$. By assuming if necessary that the two images have been obtained by removing the outer layer of a $(M+2m+2D)\times(N+2n+2D)$ image, we can ignore boundary effects.

Processing the first scan line requires computing and storing (a) $2N$ dot products of the form $\boldsymbol{w} \cdot \boldsymbol{w}$ or $\boldsymbol{w}' \cdot \boldsymbol{w}'$, $2N$ averages of the form $\bar{I}$ or $\bar{I}'$, and (c) $(2D+1)N$ dot products of the form $\boldsymbol{w} \cdot \boldsymbol{w}'$. The total storage required is $(2D+5)N$, which is certainly reasonable for, say, 1000×1000 images, and disparity ranges of $[-100, 100]$. The computation is dominated by the $\boldsymbol{w} \cdot \boldsymbol{w}'$ dot products, and its cost is on the order of $2(2m+1)(2n+1)(2D+1)N$. The incremental computations for the next scan line amount to updating all averages and dot products, with a total cost of $4(2n+1)(2D+1)N$. Assuming $M \gg m$, the overall cost of the correlation is therefore, after M updates, $4(2n+1)(2D+1)MN$ operations. Note that a naive implementation would require instead $2(2m+1)(2n+1)(2D+1)MN$ operations.

11.8. Show how a first-order expansion of the disparity function for rectified images can be used to warp the window of the right image corresponding to a rectangular region of the left one. Show how to compute correlation in this case using interpolation to estimate right-image values at the locations corresponding to the centers of the left window's pixels.

Solution Let us set up local coordinate systems whose origins are at the two points of interest—that is, the two matched points have coordinates $(0,0)$ in these coordinate systems. If $d(u,v)$ denotes the disparity function in the neighborhood of the first point, and α and β denotes its derivatives in $(0,0)$, we can write the coordinates of a match (u', v') for the point (u,v) in the first image as

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u + d(u,v) \\ v \end{pmatrix},$$

or approximating d by its first-order Taylor expansion,

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} 1+\alpha & \beta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

It follows that, to first-order, a small rectangular region in the first image maps onto a parallelogram in the second image, and that the corresponding affine transformation is completely determined by the derivatives of the disparity function.
To exploit this property in stereo matching, one can map the centers of the pixels contained in the left window onto their right images, calculate the corresponding intensity values via bilinear interpolation of neighborhing pixels in the right image, and finally computer the correlation function from these values. This is essentially the method described in Devernay and Faugeras (1994).

11.9. Show how to use the trifocal tensor to predict the tangent line along an image curve from tangent line measurements in two other pictures.

Solution Let us assume we have estimated the trifocal tensor associated with three images of a curve Γ. Let us denote by p_i the projection of a point P of Γ onto image number i $(i = 1,2,3)$. The tangent line T to Γ in P projects onto the tangent line t_i to γ_i. Given the coordinate vectors $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ of t_2 and t_2, we can predict the coordinate vector $\boldsymbol{t}_1$ of t_1 as $\boldsymbol{t}_1 = (\boldsymbol{t}_2^T \mathcal{G}_1^1 \boldsymbol{t}_3, \boldsymbol{t}_2^T \mathcal{G}_1^2 \boldsymbol{t}_3, \boldsymbol{t}_2^T \mathcal{G}_1^3 \boldsymbol{t}_3)^T$. This is another method for transfer, this time applied to lines instead of points.

Programming Assignments

11.10. Implement the rectification process.

11.11. Implement a correlation-based approach to stereopsis.

11.12. Implement a multiscale approach to stereopsis.

11.13. Implement a dynamic-programming approach to stereopsis.
11.14. Implement a trinocular approach to stereopsis.

CHAPTER 12

Affine Structure from Motion

PROBLEMS

12.1. Explain why any definition of the "addition" of two points or of the "multiplication" of a point by a scalar necessarily depends on the choice of some origin.

Solution Any such definition should be compatible with vector addition. In particular, when $R = P + Q$,we should also have $\overrightarrow{OR} = \overrightarrow{OP} + \overrightarrow{OQ}$ and $\overrightarrow{O'R} = \overrightarrow{O'P} + \overrightarrow{O'Q}$ for any choice of the origins O and O'. Subtracting the two expressions shows that $\overrightarrow{OO'} = 2\overrightarrow{OO'}$ or $O' = O$. Thus any definition of point addition would have to be relative to a fixed origin. Likewise, if $Q = \lambda P$, we should also have $\overrightarrow{OQ} = \lambda\overrightarrow{OP}$ and $\overrightarrow{O'Q} = \lambda\overrightarrow{O'P}$ for any choice of origins O and O'. This implies that $\overrightarrow{OO'} = \lambda\overrightarrow{OO'}$ or $O = O'$ when $\lambda \neq 1$. Thus any definition of point multiplication by a scalar would have to be relative to a fixed origin.

12.2. Show that the definition of a barycentric combination as

$$\sum_{i=0}^{m} \alpha_i A_i \stackrel{\text{def}}{=} A_j + \sum_{i=0, i\neq j}^{m} \alpha_i (A_i - A_j),$$

is independent of the choice of j.

Solution Let us define

$$P = A_j + \sum_{i=0, i\neq j}^{m} \alpha_i (A_i - A_j) = A_k + (A_j - A_k) + \sum_{i=0, i\neq j}^{m} \alpha_i (A_i - A_j).$$

Noting that the summation defining P can be taken over the whole $0..m$ range without changing its result and using the fact that $\sum_{i=0}^{m} \alpha_i = 1$, we can write

$$P = A_k + \sum_{i=0}^{m} \alpha_i [(A_i - A_j) + (A_j - A_k)].$$

Now, we can write $A_i = A_k + (A_i - A_k)$, but also $A_i = A_j + (A_i - A_j) = A_k + (A_j - A_k) + (A_i - A_j)$. Thus, by definition of an affine space, we must have $(A_i - A_j) + (A_j - A_k) = (A_i - A_k)$ (which is intuitively obvious). It follows that

$$P = A_k + \sum_{i=0}^{m} \alpha_i (A_i - A_k).$$

As before, omitting the term corresponding to $i = k$ does not change the result of the summation, which proves the result.

12.3. Given the two affine coordinate systems $(A) = (O_A, \boldsymbol{u}_A, \boldsymbol{v}_A, \boldsymbol{w}_A)$ and $(B) = (O_B, \boldsymbol{u}_B, \boldsymbol{v}_B, \boldsymbol{w}_B)$ for the affine space $\mathbb{E}^3$, let us define the 3×3 matrix

$$ {}^B_A\mathcal{C} = \begin{pmatrix} {}^B\boldsymbol{u}_A & {}^B\boldsymbol{v}_A & {}^B\boldsymbol{w}_A \end{pmatrix}, $$

where ${}^B\boldsymbol{a}$ denotes the coordinate vector of the vector $\boldsymbol{u}$ in the (vector) coordinate system $(\boldsymbol{u}_A, \boldsymbol{v}_A, \boldsymbol{w}_A)$. Show that

$$ {}^BP = {}^B_A\mathcal{C}\,{}^AP + {}^BO_A \quad \text{or, equivalently,} \quad \begin{pmatrix} {}^BP \\ 1 \end{pmatrix} = \begin{pmatrix} {}^B_A\mathcal{C} & {}^BO_A \\ \boldsymbol{0}^T & 1 \end{pmatrix} \begin{pmatrix} {}^AP \\ 1 \end{pmatrix}. $$

Solution The proof follows the derivation of the Euclidean change of coordinates in chapter 2. We write

$$ \overrightarrow{O_BP} = \begin{pmatrix} \boldsymbol{u}_B & \boldsymbol{v}_B & \boldsymbol{w}_B \end{pmatrix} {}^BP = \overrightarrow{O_BO_A} + \begin{pmatrix} \boldsymbol{u}_A & \boldsymbol{v}_A & \boldsymbol{w}_A \end{pmatrix} {}^AP. $$

Rewriting this equation in the coordinate frame (B) yields immediately

$$ {}^BP = {}^B_A\mathcal{C}\,{}^AP + {}^BO_A $$

since ${}^B_B\mathcal{C}$ is obviously the identity. The homogeneous form of this expression follows immediately, exactly as in the Euclidean case.

12.4. Show that the set of barycentric combinations of $m+1$ points $A_0, \ldots, A_m$ in X is indeed an affine subspace of X, and show that its dimension is at most m.

Solution Let us denote by Y the set of barycentric combinations of the points $A_0, A_1, \ldots, A_m$, pick some number j between 0 and m, and denote by U_j the vector space defined by all linear combinations of the vectors $A_i - A_j$ for $i \neq j$. It is clear that $Y = A_j + U_j$ since any barycentric combination of the points $A_0, A_1, \ldots, A_m$ is by definition in $A_j + U_j$, and any point in $A_j + U_j$ can be written as a barycentric combination of the points $A_0, A_1, \ldots, A_m$ with $\alpha_j = 1 - \sum_{i=0, i\neq j}^{m} \alpha_i$.
Thus Y is indeed an affine subspace of X, and its dimension is at most m since the vector space U_j is spanned by m vectors. This subspace is of course independent of the choice of j: This follows directly from the fact that the definition of a barycentric combination is also independent of the choice of j.

12.5. Derive the equation of a line defined by two points in $\mathbb{R}^3$. (Hint: You actually need *two* equations.)

Solution We equip $\mathbb{R}^3$ with a fixed affine coordinate system and identify points with their (non-homogeneous) coordinate vectors. According to Section 12.1.2, a necessary and sufficient for the three points $\boldsymbol{P}_1 = (x_1, y_1, z_1)^T$, $\boldsymbol{P}_2 = (x_2, y_2, z_2)^T$, and $\boldsymbol{P} = (x, y, z)^T$ to define a line (i.e., a one-dimensional affine space) is that the matrix

$$ \begin{pmatrix} x_1 & x_2 & x \\ y_1 & y_2 & y \\ z_1 & z_2 & z \\ 1 & 1 & 1 \end{pmatrix} $$

have rank 2, or equivalently, that all its 3×3 minors have zero determinant (we assume that the three points are distincts so the matrix has at least rank 2). Note that three of these determinants are

$$\begin{vmatrix} y_1 & y_2 & y \\ z_1 & z_2 & z \\ 1 & 1 & 1 \end{vmatrix} = y(z_1 - z_2) - z(y_1 - y_2) + y_1 z_2 - y_2 z_1,$$

$$\begin{vmatrix} z_1 & z_2 & z \\ x_1 & x_2 & x \\ 1 & 1 & 1 \end{vmatrix} = z(x_1 - x_2) - x(z_1 - z_2) + z_1 x_2 - z_2 x_1,$$

$$\begin{vmatrix} x_1 & x_2 & x \\ y_1 & y_2 & y \\ 1 & 1 & 1 \end{vmatrix} = x(y_1 - y_2) - y(x_1 - x_2) + x_1 y_2 - x_2 y_1,$$

i.e., the coordinates of

$$\boldsymbol{P} \times (\boldsymbol{P}_1 - \boldsymbol{P}_2) + \boldsymbol{P}_1 \times \boldsymbol{P}_2 = (\boldsymbol{P} - \boldsymbol{P}_2) \times (\boldsymbol{P}_1 - \boldsymbol{P}_2).$$

As could have been expected, writing that these three coordinates are zero is equivalent to writing that P_1, P_2, and P are collinear. Only two of the equations associated with the three coordinates of the cross product are equivalent. It is easy to see that the fourth minor is a linear combination of the other three, so the line is defined by any two of the above equations.

12.6. Show that the intersection of a plane with two parallel planes consists of two parallel lines.

Solution Consider the plane $A+U$ and the two parallel planes $B+V$ and $C+V$ in some affine space X. Here A, B, and C are points in X, and U and V are vector planes in $\vec{X}$, and we will assume from now on that U and V are distinct (otherwise the three planes are parallel). As shown in Example 12.2, the intersection of two affine subspaces $A+U$ and $B+V$ is an affine subspace associated with the vector subspace $W = U \cap V$. The intersection of two distinct planes in a vector space is a line, thus the intersection of $A+U$ and $B+V$ is a line. The same reasoning shows that the intersection of $A+U$ and $C+V$ is also a line associated with W. The two lines are parallel since they are associated with the same vector subspace W.

12.7. Show that an affine transformation $\psi : X \to Y$ between two affine subspaces X and Y associated with the vector spaces $\vec{X}$ and $\vec{Y}$ can be written as $\psi(P) = \psi(O) + \vec{\psi}(P - O)$, where O is some arbitrarily chosen origin, and $\vec{\psi} : \vec{X} \to \vec{Y}$ is a linear mapping from $\vec{X}$ onto $\vec{Y}$ that is independent of the choice of O.

Solution Let us pick some point O in X and define $\vec{\psi} : \vec{X} \to \vec{Y}$ by $\vec{\psi}(\boldsymbol{u}) = \psi(O + \boldsymbol{u}) - \psi(O)$. Clearly, $\psi(P) = \psi(O + (P - O)) = \psi(O) + \vec{\psi}(P - O)$. To show that $\vec{\psi}$ is indeed a linear mapping, let us consider two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\vec{X}$, two scalars λ and μ in $\mathbb{R}$, and the points $A = O + \boldsymbol{u}$ and $B = O + \boldsymbol{v}$. Since ψ is an affine mapping, we have

$$\begin{aligned}
\vec{\psi}(\lambda\boldsymbol{u} + \mu\boldsymbol{v}) &= \psi(O + \lambda\boldsymbol{u} + \mu\boldsymbol{v}) - \psi(O) \\
&= \psi(O + \lambda(A - O) + \mu(B - O)) - \psi(O) \\
&= \psi((1 - \lambda - \mu)O + \lambda A + \mu B) - \psi(O) \\
&= (1 - \lambda - \mu)\psi(O) + \lambda\psi(A) + \mu\psi(B) - \psi(O) \\
&= \psi(O) + \lambda(\psi(A) - \psi(O)) + \mu(\psi(B) - \psi(O)) - \psi(O) \\
&= \lambda\vec{\psi}(\boldsymbol{u}) + \mu\vec{\psi}(\boldsymbol{v}).
\end{aligned}$$

Thus $\vec{\psi}$ is indeed a linear mapping. Let us conclude by showing that it is independent of the choice of O. We define the mappings $\vec{\psi}_O$ and $\vec{\psi}_{O'}$ from $\vec{X}$ to $\vec{Y}$ by $\vec{\psi}_O(\boldsymbol{u}) = \psi(O+\boldsymbol{u}) - \psi(O)$ and $\vec{\psi}_{O'}(\boldsymbol{u}) = \psi(O'+\boldsymbol{u}) - \psi(O')$. Now,

$$\vec{\psi}_{O'}(\boldsymbol{u}) = \psi(O'+\boldsymbol{u}) - \psi(O') = \psi(O+(O'-O)+\boldsymbol{u}) - \psi(O+(O'-O)) = \vec{\psi}_O(\boldsymbol{u}),$$

thus $\vec{\psi}$ is independent of the choice of O.

12.8. Show that affine cameras (and the corresponding epipolar geometry) can be viewed as the limit of a sequence of perspective images with increasing focal length receding away from the scene.

Solution As shown in chapter 2, the projection matrix associated with a pinhole camera can be written as $\mathcal{M} = \mathcal{K}\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\end{pmatrix}$, where $\mathcal{K}$ is the matrix of intrinsic parameters, $\mathcal{R} = {}^C_W\mathcal{R}$, and $\boldsymbol{t} = {}^C O_W$. It follows that t_z, the third coordinate of $\boldsymbol{t}$, can be interpreted as the depth of the origin O_W of the world coordinate system relative to the camera. Now let us consider a camera moving away from O_W along its optical axis while zooming. Its projection matrix can be written as

$$\mathcal{M}_{\lambda,\mu} = \begin{pmatrix} \lambda\alpha & -\lambda\alpha\cot\theta & u_0 \\ 0 & \dfrac{\lambda\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & t_x \\ \mathcal{R} & t_y \\ & \mu t_z \end{pmatrix},$$

where λ and μ are the parameters controlling respectively the zoom and camera motion, with $\mathcal{M}_{1,1} = \mathcal{M}$. We can rewrite this matrix as

$$\mathcal{M}_{\lambda,\mu} = \mathcal{K}\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & t_x \\ \mathcal{R} & t_y \\ & \mu t_z \end{pmatrix} = \mathcal{K}\begin{pmatrix} \lambda\boldsymbol{r}_1^T & \lambda t_x \\ \lambda\boldsymbol{r}_2^T & \lambda t_y \\ \boldsymbol{r}_3^T & \mu t_z \end{pmatrix}.$$

Now if we choose $\mu = \lambda$ we can write

$$\mathcal{M}_{\lambda,\lambda} = \lambda\mathcal{K}\begin{pmatrix} \boldsymbol{r}_1^T & t_x \\ \boldsymbol{r}_2^T & t_y \\ \frac{1}{\lambda}\boldsymbol{r}_3^T & t_z \end{pmatrix}.$$

When $\lambda \to +\infty$, the projection becomes affine, with affine projection matrix

$$\frac{1}{t_z}\begin{pmatrix}\mathcal{K}_2\mathcal{R}_2 & \mathcal{K}_2\boldsymbol{t}_2 + \boldsymbol{p}_0\end{pmatrix},$$

where we follow the notation used in Eq. (2.19) of chapter 2.
Note that picking $\mu = \lambda$ ensures that the magnification remains constant for the fronto-parallel plane Π_0 that contains O_W. Indeed, let us denote by $(\boldsymbol{i}_C, \boldsymbol{j}_C, \boldsymbol{k}_C)$ the camera coordinate system, and consider a point $A = O_W + x\boldsymbol{i}_C + y\boldsymbol{j}_C$ in Π_0. Since $\mathcal{R} = {}^C_W\mathcal{R} = {}^W_C\mathcal{R}^T$, we have ${}^W\boldsymbol{i}_C{}^T = \boldsymbol{r}_1^T$, ${}^W\boldsymbol{j}_C{}^T = \boldsymbol{r}_2^T$, and ${}^W\boldsymbol{k}_C{}^T = \boldsymbol{r}_3^T$. It follows that

$$\mathcal{M}_{\lambda,\lambda}{}^W A = \lambda\mathcal{K}\begin{pmatrix} \boldsymbol{r}_1^T(x\boldsymbol{r}_1 + y\boldsymbol{r}_2) + t_x \\ \boldsymbol{r}_2^T(x\boldsymbol{r}_1 + y\boldsymbol{r}_2) + t_y \\ \frac{1}{\lambda}\boldsymbol{r}_3^T(x\boldsymbol{r}_1 + y\boldsymbol{r}_2) + t_z \end{pmatrix} = \lambda\mathcal{K}\begin{pmatrix} x + t_z \\ y + t_y \\ t_z \end{pmatrix}.$$

Thus the denominator involved in the perspective projection equation is equal to t_z and the same for all points in Π_0, which in turns implies that the magnification associated with Π_0 is independent of λ.

12.9. Generalize the notion of multilinearities introduced in chapter 10 to the affine case.

Solution Let us consider an affine 2×4 projection matrix $\mathcal{M}$ with rows $\mathcal{M}^1$ and $\mathcal{M}^2$. Note that we can write the projection equations, just as in chapter 10, as

$$\begin{pmatrix} u\mathcal{M}^3-\mathcal{M}^1 \\ v\mathcal{M}^3-\mathcal{M}^2 \end{pmatrix}\boldsymbol{P}=0,$$

where this time $\mathcal{M}^3=(0,0,0,1)$. We can thus construct as in that chapter the 8×4 matrix $\mathcal{Q}$, and all its 4×4 minors must, as before, have zero determinant. This yields multi-image constraints involving two, three, or four images, but since image coordinates only occur in the fourth column of $\mathcal{Q}$, these constraints are now *linear* in these coordinates (note the similarity with the affine fundamental matrix). On the other hand, the multi-image relations between lines remain multilinear in the affine case. For example, the derivation of the trifocal tensor for lines in Section 10.2.1 remains unchanged (except for the fact the third row of $\mathcal{M}$ is now equal to $(0,0,0,1)$), and yields trilinear relationships among the three lines' coordinate vectors. Likewise, the interpretation of the quadrifocal tensor in terms of lines remains valid in the affine case.

12.10. Prove Theorem 3.

Solution Let us write the singular value decomposition of $\mathcal{A}$ as $\mathcal{A}=\boldsymbol{U}\mathcal{W}\mathcal{V}^T$. Since $\mathcal{U}$ is column-orthogonal, we have

$$\mathcal{A}^T\mathcal{A}=\mathcal{V}\mathcal{W}^T\mathcal{U}^T\mathcal{U}\mathcal{W}\mathcal{V}^T=\mathcal{V}\mathcal{W}^T\mathcal{W}\mathcal{V}^T.$$

Now let $\boldsymbol{c}_i$ $(i=1,\dots,n)$ denote the columns of $\mathcal{V}$. Since $\mathcal{V}$ is orthogonal, we have

$$\begin{aligned}\mathcal{A}^T\mathcal{A}\boldsymbol{c}_i &= \mathcal{V}\mathcal{W}^T\mathcal{W}\begin{pmatrix}\boldsymbol{c}_1^T\\ \vdots\\ \boldsymbol{c}_{i-1}\\ \boldsymbol{c}_i\\ \boldsymbol{c}_{i+1}\\ \vdots\\ \boldsymbol{c}_n\end{pmatrix}\boldsymbol{c}_i=\mathcal{V}\mathrm{diag}(w_1^2,\dots,w_{i-1}^2,w_i^2,w_{i+1}^2,\dots,w_n^2)\begin{pmatrix}0\\ \vdots\\ 0\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}\\ &= \begin{pmatrix}\boldsymbol{c}_1,\dots,\boldsymbol{c}_{i-1},\boldsymbol{c}_i,\boldsymbol{c}_{i+1},\dots,\boldsymbol{c}_n\end{pmatrix}\begin{pmatrix}0\\ \vdots\\ 0\\ w_i^2\\ 0\\ \vdots\\ 0\end{pmatrix}=w_i^2\boldsymbol{c}_i.\end{aligned}$$

It follows that the vectors $\boldsymbol{c}_i$ are indeed eigenvectors of $\mathcal{A}^T\mathcal{A}$, and that the singular values are the nonnegative square roots of the corresponding eigenvalues.

12.11. Show that a calibrated paraperspective camera is an affine camera that satisfies the constraints

$$\boldsymbol{a}\cdot\boldsymbol{b}=\frac{u_rv_r}{2(1+u_r^2)}|\boldsymbol{a}|^2+\frac{u_rv_r}{2(1+v_r^2)}|\boldsymbol{b}|^2 \quad\text{and}\quad (1+v_r^2)|\boldsymbol{a}|^2=(1+u_r^2)|\boldsymbol{b}|^2,$$

where (u_r, v_r) denote the coordinates of the perspective projection of the point R.

Solution Recall from chapter 2 that the paraperspective projection matrix can be written as

$$\mathcal{M} = \frac{1}{z_r}\left(\begin{pmatrix} k & s & u_0 - u_r \\ 0 & 1 & v_0 - v_r \end{pmatrix}\mathcal{R} \quad \begin{pmatrix} k & s \\ 0 & 1 \end{pmatrix}\boldsymbol{t}_2\right).$$

For calibrated cameras, we can take $k = 1$, $s = 0$, and $u_0 = v_0 = 0$. If $\boldsymbol{r}_1^T$, $\boldsymbol{r}_2^T$, and $\boldsymbol{r}_3^T$ are the rows of the rotation matrix $\mathcal{R}$, it follows that

$$\boldsymbol{a} = \frac{1}{z_r}(\boldsymbol{r}_1 - u_r\boldsymbol{r}_3) \quad \text{and} \quad \boldsymbol{b} = \frac{1}{z_r}(\boldsymbol{r}_2 - v_r\boldsymbol{r}_3).$$

In particular, we have $|\boldsymbol{a}|^2 = (1 + u_r^2)/z_r^2$, $|\boldsymbol{b}|^2 = (1 + v_r^2)/z_r^2$, and $\boldsymbol{a} \cdot \boldsymbol{b} = u_r v_r / z_r^2$. The result immediately follows.

12.12. What do you expect the RREF of an $m \times n$ matrix with random entries to be when $m \geq n$? What do you expect it to be when $m < n$? Why?

Solution A random $m \times n$ matrix $\mathcal{A}$ usually has maximal rank. When $m > n$, this rank is n, all columns are base columns, and the $m - n$ bottom rows of the RREF of $\mathcal{A}$ are zero. When $m < n$, the rank is m, and the first m columns of $\mathcal{A}$ are normally independent. It follows that the base columns of the RREF are its first m columns; the $n - m$ rightmost columns of the RREF contain the coordinates of the corresponding columns of $\mathcal{A}$ in the basis formed by its first m columns. There are no zeros in the RREF in this case.

Programming Assignments

12.13. Implement the Koenderink–Van Doorn approach to affine shape from motion.

12.14. Implement the estimation of affine epipolar geometry from image correspondences and the estimation of scene structure from the corresponding projection matrices.

12.15. Implement the Tomasi–Kanade approach to affine shape from motion.

12.16. Add random numbers uniformly distributed in the $[0, 0.0001]$ range to the entries of the matrix $\mathcal{U}$ used to illustrate the RREF and compute its RREF (using, e.g., the `rref` routine in MATLAB); then compute again the RREF using a "robustified" version of the reduction algorithm (using, e.g., `rref` with a nonzero tolerance). Comment on the results.

CHAPTER 13

Projective Structure from Motion

PROBLEMS

13.1. Use a simple counting argument to determine the minimum number of point correspondences required to solve the projective structure-from-motion problem in the trinocular case.

Solution As shown at the beginning of this chapter, the projective structure-from-motion problem admits a finite number of solutions when $2mn \geq 11m + 3n - 15$, where m denotes the number of input pictures, and n denotes the number of point correspondences. This shows that the minimum number of point correspondences required to solve this problem in the trinocular case ($m = 3$) is given by $6n \geq 33 + 3n - 15$ or $n \geq 6$.

13.2. Show that the change of coordinates between the two projective frames $(A) = (A_0, A_1, A_2, A_3, A^*)$ and $(B) = (B_0, B_1, B_2, B_3, B^*)$ can be represented by $\rho^B P = {}^B_A\mathcal{T}^A P$, where ${}^A P$ and ${}^B P$ denote respectively the coordinate vectors of the point P in the frames (A) and (B), and ρ is an appropriate scale factor.

Solution Let us denote by $\boldsymbol{a}_0$, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, and $\boldsymbol{a}_3$ the representative vectors associated with the fundamental points of (A). Recall that these vectors are defined uniquely (up to a common scale factor) by the choice of the unit point A^*. Likewise, let us denote by $\boldsymbol{b}_0$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$ the representative vectors associated with the fundamental points of (B), and let (b) denote the corresponding vector basis of $\vec{X}$. Given a point P and a representative vector $\boldsymbol{v}$ for this point, we can write

$$\boldsymbol{v} = \lambda({}^A x_0 \boldsymbol{a}_0 + {}^A x_1 \boldsymbol{a}_1 + {}^A x_2 \boldsymbol{a}_2 + {}^A x_3 \boldsymbol{a}_3) = \mu({}^B x_0 \boldsymbol{b}_0 + {}^B x_1 \boldsymbol{b}_1 + {}^B x_2 \boldsymbol{b}_2 + {}^B x_3 \boldsymbol{b}_3),$$

or, in matrix form,

$$\lambda\begin{pmatrix}\boldsymbol{a}_0 & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3\end{pmatrix} {}^A P = \mu\begin{pmatrix}\boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3\end{pmatrix} {}^B P.$$

Rewriting this equation in the coordinate frame (b) yields immediately

$$\rho^B P = {}^B_A\mathcal{T}^A P, \quad \text{where} \quad {}^B_A\mathcal{T} = \begin{pmatrix}{}^b\boldsymbol{a}_0 & {}^b\boldsymbol{a}_1 & {}^b\boldsymbol{a}_2 & {}^b\boldsymbol{a}_3\end{pmatrix},$$

and $\rho = \mu/\lambda$, which proves the desired result. Note that the columns of ${}^B_A\mathcal{T}$ are related to the coordinate vectors ${}^B A_i$ by a priori unknown scale factors. A technique for computing these scale factors is given in Section 13.1.

13.3. Show that any two distinct lines in a projective plane intersect in exactly one point and that two parallel lines Δ and Δ' in an affine plane intersect at the point at infinity associated with their common direction $\boldsymbol{v}$ in the projective completion of this plane.

Hint: Use J_A to embed the affine plane in its projective closure, and write the vector of $\Pi \times \mathbb{R}$ associated with any point in $J_A(\Delta)$ (resp. $J_A(\Delta')$) as a linear combination of the vectors $(\overrightarrow{AB}, 1)$ and $(\overrightarrow{AB} + \boldsymbol{v}, 1)$ (resp. $(\overrightarrow{AB'}, 1)$ and $(\overrightarrow{AB'} + \boldsymbol{v}, 1)$), where B and B' are arbitrary points on Δ and Δ'.

Solution Consider two distinct lines Δ and Δ' in a projective plane, and let $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ and $(\boldsymbol{e}'_1, \boldsymbol{e}'_2)$ denote two bases for the associated two-dimensional vector spaces. The intersection of Δ and Δ' is the set of points $p(\boldsymbol{u})$, where $\boldsymbol{u} = \lambda \boldsymbol{e}_1 + \mu \boldsymbol{e}_2 = \lambda' \boldsymbol{e}'_1 + \mu' \boldsymbol{e}'_2$ for some value of the scalars $\lambda, \mu, \lambda', \mu'$. When $\boldsymbol{e}'_1$ can be written as a linear combination of the vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, we must have $\mu' = 0$ since otherwise $\boldsymbol{e}'_2$ would also be a (non trivial) linear combination of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ and the two lines would be the same. In this case, $p(\boldsymbol{e}'_1)$ is the unique intersection point of Δ and Δ'. Otherwise, the three vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}'_1$ are linearly independent, and the vector $\boldsymbol{e}'_2$ can be written in a unique manner as a linear combination of these vectors, yielding a unique solution (defined up to scale) for the scalars $\lambda, \mu, \lambda', \mu'$, and therefore a unique intersection for the lines Δ and Δ'.
Now let us consider two parallel (and thus distinct) lines Δ and Δ' with direction $\boldsymbol{v}$ in the affine plane. The intersection of their images $J_A(\Delta)$ and $J_A(\Delta')$ is determined by the solutions of the equation $\lambda(\overrightarrow{AB}, 1) + \mu(\overrightarrow{AB} + \boldsymbol{v}, 1) = \lambda'(\overrightarrow{AB'}, 1) + \mu'(\overrightarrow{AB'} + \boldsymbol{v}, 1)$. This equation can be rewritten as

$$\lambda + \mu = \lambda' + \mu' \quad \text{and} \quad (\lambda + \mu)\overrightarrow{BB'} + (\mu' - \mu)\boldsymbol{v} = \boldsymbol{0}.$$

Since the lines are not the same, the vectors $\overrightarrow{BB'}$ and $\boldsymbol{v}$ are not proportional to each other, thus we must have $\mu = \mu'$ and $\lambda + \mu = \lambda' + \mu' = 0$. Thus the two lines $J_A(\Delta)$ and $J_A(\Delta')$ intersect at the point associated with the vector

$$((\lambda + \mu)\overrightarrow{AB} + \mu\boldsymbol{v}, \lambda + \mu) = (\mu\boldsymbol{v}, 0),$$

which is the point at infinity associated with the direction $\boldsymbol{v}$.

13.4. Show that a perspective projection between two planes of $\mathbb{P}^3$ is a projective transformation.

Solution Let us consider two planes Π and Π' of $\mathbb{P}^3$ and a point O in $\mathbb{P}^3$ that does not belong to either plane. Now let us define the corresponding perspective projection $\psi : \Pi \to \Pi'$ as the mapping that associates with any point P in Π the point P' where the line passing through O and P intersects Π'. This function is bijective since any line in $\mathbb{P}^3$ that does not belong to a plane intersects this plane in exactly one point, and the inverse of ψ can be defined as the perspective projection from Π' onto Π. The function ψ obviously maps lines onto lines (the image of a line Δ in Π is the line Δ' where the plane defined by O and Δ intersects Π'). Given four points A, B, C, and D lying on the same line Δ in Π, the cross-ratio of these four points is equal to the cross-ratio of the lines Δ_A, Δ_B, Δ_C and Δ_D passing through these points and the point O. But this cross ratio is also equal to the cross ratio of the image points A', B', C', and D' that all lie on the image Δ' of Δ. Thus ψ is a projective transformation. This construction is obviously correct for the finite points of Π. It remains valid for points at infinity using the definition of the cross-ratio extended to the whole projective line.

13.5. Given an affine space X and an affine frame $(A_0, \ldots, A_n)$ for that space, what is the projective basis of $\tilde{X}$ associated with the vectors $\boldsymbol{e}_i \stackrel{\text{def}}{=} (\overrightarrow{A_0 A_i}, 0)$ $(i = 1, \ldots, n)$ and the vector $\boldsymbol{e}_{n+1} = (\boldsymbol{0}, 1)$? Are the points $J_{A_0}(A_i)$ part of that basis?

Solution The fundamental points of this projective basis are the point $p(\boldsymbol{e}_i)$ $(i = 1, \ldots, n+1)$. All but the last one lie in the hyperplane at infinity. The unit point is $p(\sum_{i=1}^{n+1} \boldsymbol{e}_i)$. The n points $J_{A_0}(A_i)$ $(i = 1, \ldots, n)$ are all finite and do not belong to the projective basis. Their coordinates are $(1, 0, \ldots, 0, 1)^T$, ..., $(0, 0, \ldots, 1, 1)^T$

13.6. In this exercise, you will show that the cross-ratio of four collinear points A, B, C, and D is equal to

$$\{A, B; C, D\} = \frac{\sin(\alpha+\beta)\sin(\beta+\gamma)}{\sin(\alpha+\beta+\gamma)\sin\beta},$$

where the angles α, β, and γ are defined as in Figure 13.2.

(a) Show that the area of a triangle PQR is

$$A(P, Q, R) = \frac{1}{2}PQ \times RH = \frac{1}{2}PQ \times PR\sin\theta,$$

where PQ denotes the distance between the two points P and Q, H is the projection of R onto the line passing through P and Q, and θ is the angle between the lines joining the point P to the points Q and R.

(b) Define the ratio of three collinear points A, B, C as

$$R(A, B, C) = \frac{\overline{AB}}{\overline{BC}}$$

for some orientation of the line supporting the three points. Show that $R(A, B, C) = A(A, B, O)/A(B, C, O)$, where O is some point not lying on this line.

(c) Conclude that the cross-ratio $\{A, B; C, D\}$ is indeed given by the formula above.

Solution

(a) The distance between the points H and R is by construction $HR = PR\sin\theta$. It is possible to construct a rectangle of dimensions $PQ \times RH$ by adding to the triangles PHR and RHQ their mirror images relative to the lines PR and RQ respectively. The area $A(P, Q, R)$ of the triangle PQR is half the area of the rectangle, i.e.,

$$A(P, Q, R) = \frac{1}{2}PQ \times RH = \frac{1}{2}PQ \times PR\sin\theta.$$

(b) Let H denote the orthogonal projection of the point O onto the line passing through the points A, B, and C. According to (a), we have $A(A, B, O) = \frac{1}{2}AB \times OH$, and $A(B, C, O) = \frac{1}{2}BC \times OH$. Thus

$$R(A, B, C) = \frac{\overline{AB}}{\overline{BC}} = \varepsilon\frac{A(A, B, O)}{A(B, C, O)},$$

where $\varepsilon = \mp 1$. Taking the convention that the area $A(P, Q, R)$ is negative when the points P, Q, and R are in clockwise order yields the desired result.

(c) By definition of the cross-ratio,

$$\{A, B; C, D\} = \frac{\overline{CA}}{\overline{CB}}\frac{\overline{DB}}{\overline{DA}} = \frac{-R(A, C, B)}{-R(A, D, B)} = \frac{R(A, C, B)}{R(A, D, B)}.$$

Now, according to (a) and (b), we have, with the same sign convention as before

$$R(A, C, B) = \frac{A(A, C, O)}{A(C, B, O)} = \frac{OA \times OC\sin(\alpha+\beta)}{-OB \times OC\sin\beta} = -\frac{OA\sin(\alpha+\beta)}{OB\sin\beta}$$

and

$$R(A,D,B) = \frac{A(A,D,O)}{A(D,B,O)} = \frac{OA \times OD \sin(\alpha+\beta+\gamma)}{-\,OB \times OD \sin(\beta+\gamma)} = -\frac{OA \sin(\alpha+\beta+\gamma)}{OB \sin(\beta+\gamma)},$$

thus

$$\{A,B;C,D\} = \frac{\sin(\alpha+\beta)\sin(\beta+\gamma)}{\sin(\alpha+\beta+\gamma)\sin\beta}.$$

13.7. Show that the homography between two epipolar pencils of lines can be written as

$$\tau \to \tau' = \frac{a\tau + b}{c\tau + d},$$

where τ and τ' are the slopes of the lines.

Solution The coordinate vectors of all lines in the pencil passing through the point $(\alpha,\beta)^T$ of the first image can be written as a linear combination of the vertical and horizontal lines going through that point, i.e., $\boldsymbol{l} = \lambda \boldsymbol{v} + \mu \boldsymbol{h}$, with $\boldsymbol{v} = (1,0,-\alpha)^T$ and $\boldsymbol{h} = (0,1,-\beta)^T$. Likewise, we can write any line in the pencil of lines passing through the point $(\alpha',\beta')^T$ of the second image as $\boldsymbol{l}' = \lambda'\boldsymbol{v}' + \mu'\boldsymbol{h}'$, with $\boldsymbol{v}' = (1,0,-\alpha')^T$ and $\boldsymbol{h}' = (0,1,-\beta')^T$. We can thus write the linear map associated with the epipolar transformation as

$$\begin{pmatrix}\lambda'\\ \mu'\end{pmatrix} = \begin{pmatrix}A & B\\ C & D\end{pmatrix}\begin{pmatrix}\lambda\\ \mu\end{pmatrix},$$

Now the slope of the line $\boldsymbol{l} = (\lambda,\mu,-\lambda\alpha-\mu\beta)^T$ is $\tau = -\lambda/\mu$, and the slope of the line $\boldsymbol{l}'$ is $\tau' = -\lambda'/\mu'$. It follows that

$$\tau' = -\frac{A\lambda + B\mu}{C\lambda + D\mu} = -\frac{-\,A\tau + B}{-\,C\tau + D} = \frac{a\tau + b}{c\tau + d},$$

where $a = -A$, $b = B$, $c = C$ and $d = -D$.

13.8. Here we revisit the three-point reconstruction problem in the context of the *homogeneous* coordinates of the point D in the projective basis formed by the tetrahedron (A,B,C,O') and the unit point O''. Note that the ordering of the reference points, and thus the ordering of the coordinates, is different from the one used earlier: This new choice is, like the previous one, made to facilitate the reconstruction.
We denote the (unknown) coordinates of the point D by (x,y,z,w), equip the first (resp. second) image plane with the triangle of reference a',b',c' (resp. a'',b'',c'') and the unit point e' (resp. e''), and denote by (x',y',z') (resp. (x'',y'',z'')) the coordinates of the point d' (resp. d'').
Hint: Drawing a diagram similar to Figure 13.3 helps.

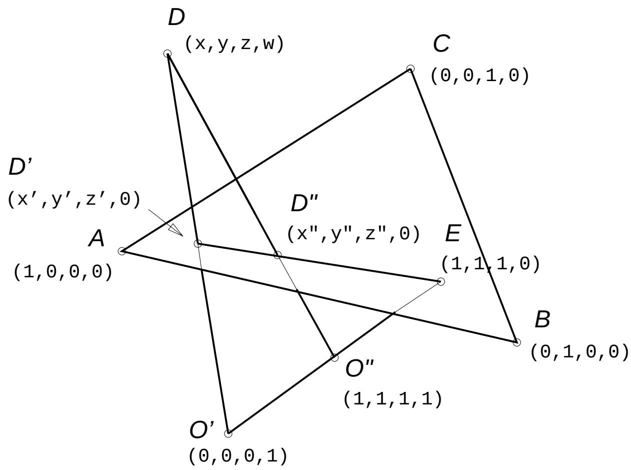
(a) What are the homogeneous projective coordinates of the points D', D'', and E where the lines $O'D$, $O''D$, and $O'O''$ intersect the plane of the triangle?

(b) Write the coordinates of D as a function of the coordinates of O' and D' (resp. O'' and D'') and some unknown parameters.

Hint: Use the fact that the points D, O', and D' are collinear.

(c) Give a method for computing these unknown parameters and the coordinates of D.

Solution The following diagram will help articulate the successive steps of the solution.



(a) Obviously, the coordinates of the points D' and D'' are simply $(x', y', z', 0)$ and $(x'', y'', z'', 0)$. The coordinates of the point E are $(1, 1, 1, 0)$.

(b) Since D lies on the line $O'D'$, we can write $\boldsymbol{D} = \lambda'\boldsymbol{O}' + \mu'\boldsymbol{D}' = \lambda''\boldsymbol{O}'' + \mu''\boldsymbol{D}''$. It remains to compute the coordinates of D as the intersection of the two rays $O'D'$ and $O''D''$.
We write $\boldsymbol{D} = \lambda'\boldsymbol{O}' + \mu'\boldsymbol{D}' = \lambda''\boldsymbol{O}'' + \mu''\boldsymbol{D}''$, which yields:

$$\begin{cases} x = \mu' x' = \lambda'' + \mu'' x'', \\ y = \mu' y' = \lambda'' + \mu'' y'', \\ z = \mu' z' = \lambda'' + \mu'' z'', \\ w = \lambda' = \lambda''. \end{cases} \tag{13.1}$$

(c) The values of μ', μ'', λ'' are found (up to some scale factor) by solving the following homogeneous system:

$$\begin{pmatrix} -x' & x'' & 1 \\ -y' & y'' & 1 \\ -z' & z'' & 1 \end{pmatrix} \begin{pmatrix} \mu' \\ \mu'' \\ \lambda'' \end{pmatrix} = 0. \tag{13.2}$$

Note that the determinant of this equation must be zero, which corresponds to D', D'', and E being collinear. In practice, (13.2) is solved through linear least-squares, and the values of x, y, z, w are then computed using (13.1).

13.9. Show that if $\tilde{\mathcal{M}} = (\mathcal{A} \quad \boldsymbol{b})$ and $\tilde{\mathcal{M}}' = (\text{Id} \quad \boldsymbol{0})$ are two projection matrices, and if $\mathcal{F}$ denotes the corresponding fundamental matrix, then $[\boldsymbol{b}_\times]\mathcal{A}$ is proportional to $\mathcal{F}$ whenever $\mathcal{F}^T\boldsymbol{b} = 0$ and

$$\mathcal{A} = -\lambda[\boldsymbol{b}_\times]\mathcal{F} + (\ \mu\boldsymbol{b} \mid \nu\boldsymbol{b} \mid \tau\boldsymbol{b}\ ).$$

Solution Suppose that $\mathcal{F}^T\boldsymbol{b} = 0$ and $\mathcal{A} = -\lambda[\boldsymbol{b}_\times]\mathcal{F} + (\ \mu\boldsymbol{b} \mid \nu\boldsymbol{b} \mid \tau\boldsymbol{b}\ )$. Since, as noted in Section 13.3, $[\boldsymbol{b}_\times]^2 = \boldsymbol{b}\boldsymbol{b}^T - |\boldsymbol{b}|^2\text{Id}$ for any vector $\boldsymbol{b}$, we have

$$\mathcal{A} = -\lambda\boldsymbol{b}\boldsymbol{b}^T\mathcal{F} + \lambda|\boldsymbol{b}|^2\mathcal{I}\lceil\mathcal{F} + [\boldsymbol{b}_\times](\ \mu\boldsymbol{b} \mid \nu\boldsymbol{b} \mid \tau\boldsymbol{b}\ ) = \lambda|\boldsymbol{b}|^2\mathcal{F}.$$

This shows that $[\boldsymbol{b}_\times]$ is indeed proportional to $\mathcal{F}$ and there exists a four-parameter family of solutions for the matrix $\tilde{\mathcal{M}}$ defined (up to scale) by the parameters λ, μ, ν, and τ.

13.10. We derive in this exercise a method for computing a minimal parameterization of the fundamental matrix and estimating the corresponding projection matrices. This is similar in spirit to the technique presented in Section 12.2.2 of chapter 12 in the affine case.

(a) Show that two projection matrices $\mathcal{M}$ and $\mathcal{M}'$ can always be reduced to the following canonical forms by an appropriate projective transformation:

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{M}}' = \begin{pmatrix} \boldsymbol{a}_1^T & b_1 \\ \boldsymbol{a}_2^T & b_2 \\ \boldsymbol{0}^T & 1 \end{pmatrix}.$$

Note: For simplicity, you can assume that all the matrices involved in your solution are nonsingular.

(b) Note that applying this transformation to the projection matrices amounts to applying the inverse transformation to every scene point P. Let us denote by $\tilde{\boldsymbol{P}} = (x, y, z)^T$ the position of the transformed point $\tilde{P}$ in the world coordinate system and by $\boldsymbol{p} = (u, v, 1)^T$ and $\boldsymbol{p'} = (u', v', 1)^T$ the homogeneous coordinate vectors of its images. Show that

$$(u' - b_1)(\boldsymbol{a}_2 \cdot \boldsymbol{p}) = (v' - b_2)(\boldsymbol{a}_1 \cdot \boldsymbol{p}).$$

(c) Derive from this equation an eight-parameter parameterization of the fundamental matrix, and use the fact that $\mathcal{F}$ is only defined up to a scale factor to construct a minimal seven-parameter parameterization.

(d) Use this parameterization to derive an algorithm for estimating $\mathcal{F}$ from at least seven point correspondences and for estimating the projective shape of the scene.

Solution

(a) Let $\boldsymbol{m}_i^T$ and $\boldsymbol{m}_i'^T$ $(i = 1, 2, 3)$ denote the rows of the matrices $\mathcal{M}$ and $\mathcal{M}'$. We can define the 4×4 matrix

$$\mathcal{N} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \\ \boldsymbol{m}_3'^T \end{pmatrix}$$

and choose $\mathcal{Q} = \mathcal{N}^{-1}$ when $\mathcal{N}$ is not singular.

(b) We can write the corresponding projection equations as

$$\begin{cases} z\boldsymbol{p} = \tilde{\boldsymbol{P}}, \\ z'\boldsymbol{p}' = \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{0}^T \end{pmatrix} \tilde{\boldsymbol{P}} + \begin{pmatrix} b_1 \\ b_2 \\ 1 \end{pmatrix} = z \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ 0 \end{pmatrix} \tilde{\boldsymbol{p}} + \begin{pmatrix} b_1 \\ b_2 \\ 1 \end{pmatrix}. \end{cases}$$

It follows that $\boldsymbol{P} = z\boldsymbol{p}$, $z' = 1$, and

$$\begin{cases} u' = z\boldsymbol{a}_1 \cdot \boldsymbol{p} + b_1, \\ v' = z\boldsymbol{a}_2 \cdot \boldsymbol{p} + b_2. \end{cases} \tag{13.3}$$

Eliminating z among these equations yields

$$(u' - b_1)(\boldsymbol{a}_2 \cdot \boldsymbol{p}) = (v' - b_2)(\boldsymbol{a}_1 \cdot \boldsymbol{p}).$$

(c) The above equation is easily rewritten in the familiar form $\boldsymbol{p}\mathcal{F}\boldsymbol{p}' = 0$ of the epipolar constraint, the fundamental matrix being written in this case as

$$\mathcal{F} = \begin{pmatrix} \boldsymbol{a}_2 & -\boldsymbol{a}_1 & b_2\boldsymbol{a}_1 - b_1\boldsymbol{a}_2 \end{pmatrix}.$$

This construction provides a parameterization of the fundamental matrix by 8 independent coefficients (the components of the vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ and the two scalars b_1 and b_2) and guarantees that $\mathcal{F}$ is singular. Since the fundamental matrix is only defined up to a scale factor, one of the coordinates of, say, the vector $\boldsymbol{a}_1$ can arbitrarily be set to 1, yielding a minimal seven-parameter parameterization.

(d) Using this parameterization, the matrix $\mathcal{F}$ can be estimated from at least 7 point matches using non-linear least-squares. Once its parameters are known, we can reconstruct every scene point as $\tilde{\boldsymbol{P}} = z\boldsymbol{p}$, where z is the least-squares solution of (13.3), i.e.,

$$z = -\frac{(\boldsymbol{a}_1 \cdot \boldsymbol{p})(b_1 - u') + (\boldsymbol{a}_2 \cdot \boldsymbol{p})(b_2 - v')}{(\boldsymbol{a}_1 \cdot \boldsymbol{p})^2 + (\boldsymbol{a}_2 \cdot \boldsymbol{p})^2}.$$

13.11. We show in this exercise that when two cameras are (internally) calibrated so the essential matrix $\mathcal{E}$ can be estimated from point correspondences, it is possible the recover the rotation $\mathcal{R}$ and translation $\boldsymbol{t}$ such $\mathcal{E} = [\boldsymbol{t}_\times]\mathcal{R}$ without solving first the projective structure-from-motion problem. (This exercise is courtesy of Andrew Zisserman.)

(a) Since the structure of a scene can only be determined up to a similitude, the translation $\boldsymbol{t}$ can only be recovered up to scale. Use this and the fact that $\mathcal{E}^T\boldsymbol{t} = 0$ to show that the SVD of the essential matrix can be written as

$$\mathcal{E} = \mathcal{U}\ \mathrm{diag}(1, 1, 0)\mathcal{V}^\top,$$

and conclude that $\boldsymbol{t}$ can be taken equal to the third column vector of $\mathcal{U}$.

(b) Show that the two matrices

$$\mathcal{R}_1 = \mathcal{U}\mathcal{W}\mathcal{V}^\top \quad \text{and} \quad \mathcal{R}_2 = \mathcal{U}\mathcal{W}^\top\mathcal{V}^\top$$

satisfy (up to an irrelevant sign change) $\mathcal{E} = [\boldsymbol{t}]_\times\mathcal{R}$, where

$$\mathcal{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Solution

(a) Since an essential matrix is singular with two equal nonzero singular values (see chapter 10), and $\mathcal{E}$ and $\boldsymbol{t}$ are only defined up to scale, we can always take the two nonzero singular values equal to 1, and write the SVD of $\mathcal{E}$ as

$$\mathcal{E} = \mathcal{U}\ \mathrm{diag}(1, 1, 0)\mathcal{V}^\top.$$

Writing $\mathcal{E}^T \boldsymbol{t} = \mathbf{0}$ now yields

$$\mathbf{0} = \mathcal{V}\mathrm{diag}(1,1,0)\mathcal{U}^T \boldsymbol{t} = \mathcal{V}\begin{pmatrix} \boldsymbol{u}_1 \cdot \boldsymbol{t} \\ \boldsymbol{u}_2 \cdot \boldsymbol{t} \\ 0 \end{pmatrix},$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the first two columns of $\mathcal{U}$. Since $\mathcal{V}$ is an orthogonal matrix, it is nonsingular and we must have $\boldsymbol{u}_1 \cdot \boldsymbol{t} = \boldsymbol{u}_2 \cdot \boldsymbol{t} = 0$. Thus $\boldsymbol{t}$ must be parallel to the third column of the orthogonal matrix $\mathcal{U}$. Since, once again, $\mathcal{E}$ is only defined up to scale, we can take $\boldsymbol{t}$ to be the third column (a unit vector) of $\mathcal{U}$.

(b) First note that we can always assume that the orthogonal matrices $\mathcal{U}$ and $\mathcal{V}$ are rotation matrices. Indeed, since the third singular value of $\mathcal{E}$ is zero, we can always replace the third column of either matrix by its opposite to make the corresponding determinant positive. The resulting decomposition of $\mathcal{E}$ is still a valid SVD. Since the matrices $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$ (and their transposes) are rotation matrices, it follows that both $\mathcal{R}_1$ and $\mathcal{R}_2$ are also rotation matrices. Now since $\boldsymbol{t}$ is the third column of $\mathcal{U}$, we have $\boldsymbol{t} \times \boldsymbol{u}_1 = \boldsymbol{u}_2$ and $\boldsymbol{t} \times \boldsymbol{u}_2 = -\boldsymbol{u}_1$. In particular,

$$\begin{aligned} [\boldsymbol{t}_\times]\mathcal{R}_1 &= \begin{pmatrix} \boldsymbol{u}_2 & -\boldsymbol{u}_1 & \mathbf{0} \end{pmatrix} \mathcal{W}\mathcal{V}^T = -\begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \mathbf{0} \end{pmatrix} \mathcal{V}^T \\ &= -\mathcal{U}\mathrm{diag}(1,1,0)\mathcal{V}^T = -\mathcal{E}. \end{aligned}$$

Likewise, it is easy to show that $[\boldsymbol{t}_\times]\mathcal{R}_2 = \mathcal{E}$. Since $\mathcal{E}$ is only defined up to scale, both solutions are valid essential matrices.

Programming Assignments

13.12. Implement the geometric approach to projective scene estimation introduced in Section 13.2.1

13.13. Implement the algebraic approach to projective scene estimation introduced in Section 13.2.2.

13.14. Implement the factorization approach to projective scene estimation introduced in Section 13.4.1.

CHAPTER 14

Segmentation by Clustering

PROBLEMS

14.1. We wish to cluster a set of pixels using color and texture differences. The objective function

$$\Phi(\text{clusters}, \text{data}) = \sum_{i \in \text{clusters}} \left\{ \sum_{j \in i\text{`th cluster}} (\boldsymbol{x}_j - \boldsymbol{c}_i)^T (\boldsymbol{x}_j - \boldsymbol{c}_i) \right\}$$

used in Section 14.4.2 may be inappropriate — for example, color differences could be too strongly weighted if color and texture are measured on different scales.

(a) Extend the description of the k-means algorithm to deal with the case of an objective function of the form

$$\Phi(\text{clusters}, \text{data}) = \sum_{i \in \text{clusters}} \left\{ \sum_{j \in i\text{`th cluster}} (\boldsymbol{x}_j - \boldsymbol{c}_i)^T \mathcal{S} (\boldsymbol{x}_j - \boldsymbol{c}_i) \right\},$$

where $\mathcal{S}$ is an a symmetric, positive definite matrix.

(b) For the simpler objective function, we had to ensure that each cluster contained at least one element (otherwise we can't compute the cluster center). How many elements must a cluster contain for the more complicated objective function?

(c) As we remarked in Section 14.4.2, there is no guarantee that k-means gets to a global minimum of the objective function; show that it must always get to a local minimum.

(d) Sketch two possible local minima for a k-means clustering method clustering data points described by a two-dimensional feature vector. Use an example with only two clusters for simplicity. You shouldn't need many data points. You should do this exercise for both objective functions.

Solution

(a) Estimate the covariance matrix for the cluster and use its inverse.

(b) $O(d^2)$ where d is the dimension of the feature vector.

(c) The value is bounded below and it goes down at each step unless it is already at a minimum with respect to that step.

(d) Do this with a symmetry; equilateral triangle with two cluster centers is the easiest.

14.2. Read Shi and Malik (2000) and follow the proof that the normalized cut criterion leads to the integer programming problem given in the text. Why does the normalized affinity matrix have a null space? Give a vector in its kernel.

Solution Read the paper.

14.3. Show that choosing a *real* vector that maximises the expression

$$\frac{\boldsymbol{y}^T(\mathcal{D}-\mathcal{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathcal{D}\boldsymbol{y}}$$

is the same as solving the eigenvalue problem

$$\mathcal{D}^{-1/2}\mathcal{W}\mathcal{W}\boldsymbol{z} = \mu\boldsymbol{z},$$

where $\boldsymbol{z} = \mathcal{D}^{-1/2}\boldsymbol{y}$.

Solution DAF suggests not setting this as an exercise, because he got it wrong (sorry!). The correct form would be: Show that choosing a *real* vector that maximises the expression

$$\frac{\boldsymbol{y}^T(\mathcal{D}-\mathcal{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathcal{D}\boldsymbol{y}}$$

is the same as solving the eigenvalue problem

$$\mathcal{D}^{-1/2}\mathcal{W}\mathcal{D}^{-1/2}\boldsymbol{z} = \mu\boldsymbol{z},$$

where $\boldsymbol{z} = \mathcal{D}^{1/2}\boldsymbol{y}$. Of course, this requires that $\mathcal{D}$ have full rank, in which case one could also solve

$$\mathcal{D}^{-1}\mathcal{W}\boldsymbol{y} = \lambda\boldsymbol{y}$$

or simply the generalized eigenvalue problem,

$$\mathcal{W}\boldsymbol{y} - \lambda\mathcal{D}\boldsymbol{y} = \boldsymbol{0}$$

which Matlab will happily deal with.

14.4. This exercise explores using normalized cuts to obtain more than two clusters. One strategy is to construct a new graph for each component separately and call the algorithm recursively. You should notice a strong similarity between this approach and classical divisive clustering algorithms. The other strategy is to look at eigenvectors corresponding to smaller eigenvalues.

(a) Explain why these strategies are not equivalent.

(b) Now assume that we have a graph that has two connected components. Describe the eigenvector corresponding to the largest eigenvalue.

(c) Now describe the eigenvector corresponding to the second largest eigenvalue.

(d) Turn this information into an argument that the two strategies for generating more clusters should yield quite similar results under appropriate conditions; what are appropriate conditions?

Solution

(a) They would be equivalent if the matrix actually was block diagonal.

(b) It has zeros in the entries corresponding to one connected component.

(c) Could be anything; but there is another eigenvector which has zeros in the entries corresponding to the other connected component. This doesn't have to correspond to the second eigenvalue.

(d) Basically, if the graph is very close to block diagonal, the eigenvectors split into a family corresponding to the eigenvectors of the first block and the eigenvectors of the second block, which implies that for a graph that is close enough to block diagonal (but what does this mean formally — we'll duck this bullet) the two strategies will be the same.

Programming Assignments

14.5. Build a background subtraction algorithm using a moving average and experiment with the filter.

14.6. Build a shot boundary detection system using any two techniques that appeal, and compare performance on different runs of video.

14.7. Implement a segmenter that uses k-means to form segments based on color and position. Describe the effect of different choices of the number of segments and investigate the effects of different local minima.

CHAPTER 15

Segmentation by Fitting a Model

PROBLEMS

15.1. Prove the simple, but extremely useful, result that the perpendicular distance from a point (u, v) to a line (a, b, c) is given by abs$(au + bv + c)$ *if* $a^2 + b^2 = 1$.

Solution Work with the squared distance; choose a point (x, y) on the line, and we now wish to minimize $(u - x)^2 + (v - y)^2$ subject to $ax + by + c = 0$. This gives

$$2\begin{pmatrix} (u-x) \\ (v-y) \end{pmatrix} + \lambda \begin{pmatrix} a \\ b \end{pmatrix} = 0,$$

which means that $((u - x), (v - y))$ is parallel to the line's normal, so $(u, v) = (x, y) + \lambda(a, b)$. Now if $a^2 + b^2 = 1$, $abs(\lambda)$ would be the distance, because (a, b) is a unit vector. But $ax + by + c = 0$, so $au + bv + c = -\lambda(a^2 + b^2) = -\lambda$ and we are done.

15.2. Derive the eigenvalue problem

$$\begin{pmatrix} \overline{x^2} - \overline{x}\,\overline{x} & \overline{xy} - \overline{x}\,\overline{y} \\ \overline{xy} - \overline{x}\,\overline{y} & \overline{y^2} - \overline{y}\,\overline{y} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \mu \begin{pmatrix} a \\ b \end{pmatrix}$$

from the generative model for total least squares. This is a simple exercise — maximum likelihood and a little manipulation will do it — but worth doing right and remembering; the technique is extremely useful.

Solution We wish to minimise $\sum_i (ax_i + by_i + c)^2$ subject to $a^2 + b^2 = 1$. This yields

$$\begin{pmatrix} \overline{x^2} & \overline{xy} & \overline{x} \\ \overline{y^2} & \overline{xy} & \overline{y} \\ \overline{x} & \overline{y} & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \lambda \begin{pmatrix} a \\ b \\ 0 \end{pmatrix},$$

where λ is the Lagrange multiplier. Now substitute back the third row (which is $\overline{x}a + \overline{y}b + c = 0$) to get the result.

15.3. How do we get a curve of edge points from an edge detector that returns orientation? Give a recursive algorithm.

15.4. A slightly more stable variation of incremental fitting cuts the first few pixels and the last few pixels from the line point list when fitting the line because these pixels may have come from a corner

(a) Why would this lead to an improvement?

(b) How should one decide how many pixels to omit?

Solution

(a) The first and last few are respectively the end of one corner and the beginning of the next, and tend to bias the fit.

(b) Experiment, though if you knew a lot about the edge detector and the lens you might be able to derive an estimate.

15.5. A conic section is given by $ax^2 + bxy + cy^2 + dx + ey + f = 0$.

(a) Given a data point (d_x, d_y), show that the nearest point on the conic (u, v) satisfies two equations:

$$\begin{aligned} au^2 + buv + cv^2 + du + ev + f &= 0 \\ &\text{and} \\ 2(a-c)uv - (2ad_y + e)u + (2cd_x + d)v + (ed_x - dd_y) &= 0. \end{aligned}$$

(b) These are two quadratic equations. Write $\boldsymbol{u}$ for the vector $(u, v, 1)$. Now show that we can write these equations as $\boldsymbol{u}^T \mathcal{M}_1 \boldsymbol{u} = 0$ and $\boldsymbol{u}^T \mathcal{M}_2 \boldsymbol{u} = 0$, for $\mathcal{M}_1$ and $\mathcal{M}_2$ symmetric matrices.

(c) Show that there is a transformation $\mathcal{T}$, such that $\mathcal{T}^T \mathcal{M}_1 \mathcal{T} = Id \; and \; \mathcal{T}^T \mathcal{M}_2 \mathcal{T}$ is diagonal.

(d) Now show how to use this transformation to obtain a set of solutions to the equations; in particular, show that there can be up to four real solutions.

(e) Show that there are four, two, or zero real solutions to these equations.

(f) Sketch an ellipse and indicate the points for which there are four or two solutions.

Solution All this is straightforward algebra, except for (c) which gives a lot of people trouble. M_1 is symmetric, so can be reduced to a diagonal form by the eigenvector matrix and to the identity using the square roots of the eigenvalues. Now any rotation matrix fixes the identity; so I can use the eigenvector matrix of M_2 to diagonalize M_2 while fixing M_1 at the identity.

15.6. Show that the curve

$$\left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$$

is a circular arc (the length of the arc depending on the interval for which the parameter is defined).

(a) Write out the equation in t for the closest point on this arc to some data point (d_x, d_y). What is the degree of this equation? How many solutions in t could there be?

(b) Now substitute $s^3 = t$ in the parametric equation, and write out the equation for the closest point on this arc to the same data point. What is the degree of the equation? Why is it so high? What conclusions can you draw?

Solution Do this by showing that

$$\left(\frac{1-t^2}{1+t^2}\right)^2 + \left(\frac{2t}{1+t^2}\right)^2 = 1.$$

(a) The normal is

$$\left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right),$$

so our equation is

$$(x - \frac{1-t^2}{1+t^2})\frac{2t}{1+t^2} + (y - \frac{2t}{1+t^2})(-\frac{1-t^2}{1+t^2}) = 0,$$

and if we clear denominators by multiplying both sides by $(1+t^2)^2$, the highest degree term in t will have degree 4, so the answer is in principle 4. But if you expand the sum out, you'll find that the degree 4 and degree 3 terms cancel, and you'll have a polynomial of degree 2.

(b) It will have degree 6 in s; this is because the parametrisation allows each point on the curve to have three different parameter values (the s value for each t is $t^{(1/3)}$, and every number has three cube roots; it is very difficult in practice to limit this sort of calculation to real values only).

15.7. Show that the viewing cone for a cone is a family of planes, all of which pass through the focal point and the vertex of the cone. Now show the outline of a cone consists of a set of lines passing through a vertex. You should be able to do this by a simple argument without any need for calculations.

Solution The viewing cone for a surface consists of all rays through the focal point and tangent to the surface. Construct a line through the focal point and the vertex of the cone. Now construct any plane through the focal point that does not pass through the vertex of the cone. This second plane slices the cone in some curve. Construct the set of tangents to this curve that pass through the focal point (which is on the plane by construction). Any plane that contains the first line and one of these tangents is tangent to the cone, and the set of such planes exhausts the planes tangent to the cone and passing through the focal point. The outline is obtained by slicing this set of planes with another plane not lying on their shared line, and so must be a set of lines passing through some common point.

Programming Assignments

15.8. Implement an incremental line fitter. Determine how significant a difference results if you leave out the first few pixels and the last few pixels from the line point list (put some care into building this, as it's a useful piece of software to have lying around in our experience).

15.9. Implement a hough transform line finder.

15.10. Count lines with an HT line finder - how well does it work?

CHAPTER 16

Segmentation and Fitting using Probabilistic Methods

PROBLEMS

16.1. Derive the expressions of Section 16.1 for segmentation. One possible modification is to use the new mean in the estimate of the covariance matrices. Perform an experiment to determine whether this makes any difference in practice.

Solution The exercises here can be done in too many different ways to make model solutions helpful. Jeff Bilmes' document, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," which can be found on CiteSeer, gives a very detailed derivation of EM for Gaussian mixture models (which is the first problem).

16.2. Supply the details for the case of using EM for background subtraction. Would it help to have a more sophisticated foreground model than uniform random noise?

16.3. Describe using leave-one-out cross-validation for selecting the number of segments.

Programming Assignments

16.4. Build an EM background subtraction program. Is it practical to insert a dither term to overcome the difficulty with high spatial frequencies illustrated in Figure 14.11?

16.5. Build an EM segmenter that uses color and position (ideally, use texture too) to segment images; use a model selection term to determine how many segments there should be. How significant a phenomenon is the effect of local minima?

16.6. Build an EM line fitter that works for a fixed number of lines. Investigate the effects of local minima. One way to avoid being distracted by local minima is to start from many different start points and then look at the best fit obtained from that set. How successful is this? How many local minima do you have to search to obtain a good fit for a typical data set? Can you improve things using a Hough transform?

16.7. Expand your EM line fitter to incorporate a model selection term so that the fitter can determine how many lines fit a dataset. Compare the choice of AIC and BIC.

16.8. Insert a noise term in your EM line fitter, so that it is able to perform robust fits. What is the effect on the number of local minima? Notice that, if there is a low probability of a point arising from noise, most points will be allocated to lines, but the fits will often be quite poor If there is a high probability of a point arising from noise, points will be allocated to lines only if they fit well. What is the effect of this parameter on the number of local minima?

16.9. Construct a RANSAC fitter that can fit an arbitrary (but known) number of lines to a given data set. What is involved in extending your fitter to determine the best number of lines?

C H A P T E R 17

Tracking with Linear Dynamic Models

PROBLEMS

17.1. Assume we have a model $\boldsymbol{x}_i = \mathcal{D}_i \boldsymbol{x}_{i-1}$ and $y_i = \boldsymbol{M}_i^T \boldsymbol{x}_i$. Here the measurement y_i is a one-dimensional vector (i.e., a single number) for each i and $\boldsymbol{x}_i$ is a k-dimensional vector. We say model is *observable* if the state can be reconstructed from any sequence of k measurements.

(a) Show that this requirement is equivalent to the requirement that the matrix

$$\left[\boldsymbol{M}_i \mathcal{D}_i^T \boldsymbol{M}_{i+1} \mathcal{D}_i^T \mathcal{D}_{i+1}^T \boldsymbol{M}_{i+2} \ldots \mathcal{D}_i^T \ldots \mathcal{D}_{i+k-2}^T \boldsymbol{M}_{i+k-1}\right]$$

has full rank.

(b) The point drifting in 3D, where $\mathcal{M}_{3k} = (0,0,1)$, $\mathcal{M}_{3k+1} = (0,1,0)$, and $\mathcal{M}_{3k+2} = (1,0,0)$ is observable.

(c) A point moving with constant velocity in any dimension, with the observation matrix reporting position only, is observable.

(d) A point moving with constant acceleration in any dimension, with the observation matrix reporting position only, is observable.

Solution There is a typo in the indices here, for which DAF apologizes. A sequence of k measurements is $(y_i, y_{i+1}, \ldots, y_{i+k-1})$. This sequence is

$$(\boldsymbol{M}_i^T \boldsymbol{x}_i, \boldsymbol{M}_{i+1}^T \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{M}_{i+k-1}^T \boldsymbol{x}_{i+k-1}).$$

Now $\boldsymbol{x}_{i+1} = \mathcal{D}_{i+1} \boldsymbol{x}_i$ etc., so the sequence is

$$(\boldsymbol{M}_i^T \boldsymbol{x}_i, \boldsymbol{M}_{i+1}^T \mathcal{D}_i \boldsymbol{x}_i, \ldots \boldsymbol{M}_{i+k-1}^T \mathcal{D}_{i+k-2} \mathcal{D}_{i+k-3} \ldots \mathcal{D}_i \boldsymbol{x}_i)$$

, which is

$$\begin{bmatrix} \boldsymbol{M}_i^T \\ \boldsymbol{M}_{i+1}^T \mathcal{D}_i \\ \ldots \\ \boldsymbol{M}_{i+k-1}^T \mathcal{D}_{i+k-2} \mathcal{D}_{i+k-3} \ldots \mathcal{D}_i \end{bmatrix} \boldsymbol{x}_i,$$

and if the rank of this matrix (which is the transpose of the one given, except for the index typo) is k, we are ok. The rest are calculations.

17.2. A point on the line is moving under the drift dynamic model. In particular, we have $x_i \sim N(x_{i-1}, 1)$. It starts at $x_0 = 0$.

(a) What is its average velocity? (Remember, velocity is *signed*.)

Solution 0.

(b) What is its average speed? (Remember, speed is *unsigned*.)

Solution This depends (a) on the timestep and (b) on the number of steps you allow before measuring the speed (sorry - DAF). But the average of 1-step speeds is 1 if the timestep is 1.

(c) How many steps, on average, before its distance from the start point is greater than two (i.e., what is the expected number of steps, etc.?)

Solution This is finicky and should not have been set — don't use it; sorry – DAF.

(d) How many steps, on average, before its distance from the start point is greater than ten (i.e., what is the expected number of steps, etc.)?

Solution This is finicky and should not have been set — don't use it; sorry – DAF.

(e) (This one requires some thought.) Assume we have two nonintersecting intervals, one of length 1 and one of length 2; what is the limit of the ratio (average percentage of time spent in interval one)/ (average percentage of time spent in interval two) as the number of steps becomes infinite?

Solution This is finicky and should not have been set — don't use it; sorry – DAF.

(f) You probably guessed the ratio in the previous question; now run a simulation and see how long it takes for this ratio to look like the right answer.

Solution This is finicky and should not have been set — sorry, DAF. The answer is 1/2, and a simulation will produce it, but will take quite a long time to do so.

17.3. We said that

$$g(x; a, b)g(x; c, d) = g(x; \frac{ad + cb}{b + d}, \frac{bd}{b + d})f(a, b, c, d).$$

Show that this is true. The easiest way to do this is to take logs and rearrange the fractions.

17.4. Assume that we have the dynamics

$$x_i \sim N(d_i x_{i-1}, \sigma_{d_i}^2);$$

$$y_i \sim N(m_i x_i, \sigma_{m_i}^2).$$

(a) $P(x_i|x_{i-1})$ is a normal density with mean $d_i x_{i-1}$ and variance $\sigma_{d_i}^2$. What is $P(x_{i-1}|x_i)$?

(b) Now show how we can obtain a representation of $P(\boldsymbol{x}_i|\boldsymbol{y}_{i+1}, \ldots, \boldsymbol{y}_N)$ using a Kalman filter.

Solution

(a) We have $x_i = d_i x_{i-1} + \zeta$, where ζ is Gaussian noise with zero mean and variance $\sigma_{d_i}^2$. This means that $x_{i-1} = (1/d_i)(x_i - \zeta) = x_i/d_i + \xi$, where ξ is Gaussian noise with zero mean and variance $\sigma_{d_i}^2/d_i^2$.

(b) Run time backwards.

Programming Assignments

17.5. Implement a 2D Kalman filter tracker to track something in a simple video sequence. We suggest that you use a background subtraction process and track the foreground blob. The state space should probably involve the position of the blob, its velocity, its orientation — which you can get by computing the matrix of second moments — and its angular velocity.

17.6. If one has an estimate of the background, a Kalman filter can improve background subtraction by tracking illumination variations and camera gain changes. Implement a Kalman filter that does this; how substantial an improvement does this offer? Notice that a reasonable model of illumination variation has the background multiplied by a noise term that is near one — you can turn this into linear dynamics by taking logs.

CHAPTER 18

Model-Based Vision

PROBLEMS

18.1. Assume that we are viewing objects in a calibrated perspective camera and wish to use a pose consistency algorithm for recognition.

(a) Show that three points is a frame group.
(b) Show that a line and a point is *not* a frame group.
(c) Explain why it is a good idea to have frame groups composed of different types of feature.
(d) Is a circle and a point not on its axis a frame group?

Solution

(a) We have a calibrated perspective camera, so in the camera frame we can construct the three rays through the focal point corresponding to each image point. We must now slice these three rays with some plane to get a prescribed triangle (the three points on the object). If there is only a discrete set of ways of doing this, we have a frame group, because we can recover the rotation and translation of the camera from any such plane. Now choose a point along ray 1 to be the first object point. There are at most two possible points on ray 2 that could be the second object point — see this by thinking about the 12 edge of the triangle as a link of fixed length, and swinging this around the first point; it forms a sphere, which can intersect a line in at most two points. Choose one of these points. Now we have fixed one edge of our triangle in space — can we get the third point on the object triangle to intersect the third image ray? In general, no, because we can only rotate the triangle about the 12 edge, which means the third point describes a circle; but a circle will not in general intersect a ray in space, so we have to choose a special point for along ray 1 to be the object point. It follows that only a discrete set of choices are possible, and we are done.

(b) We use a version of the previous argument. We have a calibrated perspective camera, and so can construct in the camera frame the plane and ray corresponding respectively to the image line and image point. Now choose a line on the plane to be the object line. Can we find a solution for the object point? The object point could lie anywhere on a cylinder whose axis is the chosen line and whose radius is the distance from line to point. There are now two general cases — either there is no solution, or there are two (where the ray intersects the cylinder). But for most lines where there are two solutions, slightly moving the line results in another line for which there are two solutions, so there is a continuous family of available solutions, meaning it can't be a frame group.

(c) Correspondence search is easier.

(d) Yes, for a calibrated perspective camera.

18.2. We have a set of plane points $\boldsymbol{P}_j$; these are subject to a plane affine transformation. Show that

$$\frac{\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_k\right]}{\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_l\right]}$$

is an affine invariant (as long as no two of i, j, k, and l are the same and no three of these points are collinear).

Solution Write $\boldsymbol{Q}_i = \mathcal{M}\boldsymbol{P}_i$ for the affine transform of point $\boldsymbol{P}_i$. Now

$$\frac{\det\left[\boldsymbol{Q}_i\boldsymbol{Q}_j\boldsymbol{Q}_k\right]}{\det\left[\boldsymbol{Q}_i\boldsymbol{Q}_j\boldsymbol{Q}_l\right]} = \frac{\det(\mathcal{M}\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_k\right])}{\det(\mathcal{M}\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_l\right])} = \frac{\det(\mathcal{M})\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_k\right]}{\det(\mathcal{M})\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_l\right]} = \frac{\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_k\right]}{\det\left[\boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_l\right]}$$

18.3. Use the result of the previous exercise to construct an affine invariant for:
(a) four lines,
(b) three coplanar points,
(c) a line and two points (these last two will take some thought).

Solution

(a) Take the intersection of lines 1 and 2 as $\boldsymbol{P}_i$, etc.

(b) Typo! can't be done; sorry - DAF.

(c) Construct the line joining the two points; these points, with the intersection between the lines, give three collinear points. The ratio of their lengths is an affine invariant. Easiest proof: an affine transformation of the plane restricted to this line is an affine transformation of the line. But this involves only scaling and translation, and the ratio of lengths is invariant to both.

18.4. In chamfer matching at any step, a pixel can be updated if the distances from some or all of its neighbors to an edge are known. Borgefors counts the distance from a pixel to a vertical or horizontal neighbor as 3 and to a diagonal neighbor as 4 to ensure the pixel values are integers. Why does this mean $\sqrt{2}$ is approximated as 4/3? Would a better approximation be a good idea?

18.5. One way to improve pose estimates is to take a verification score and then optimize it as a function of pose. We said that this optimization could be hard particularly if the test to tell whether a backprojected curve was close to an edge point was a threshold on distance. Why would this lead to a hard optimization problem?

Solution Because the error would not be differentiable — as the backprojected outline moved, some points would start or stop contributing.

18.6. We said that for an uncalibrated affine camera viewing a set of plane points, the effect of the camera can be written as an unknown plane affine transformation. Prove this. What if the camera is an uncalibrated perspective camera viewing a set of plane points?

18.7. Prepare a summary of methods for registration in medical imaging other than the geometric hashing idea we discussed. You should keep practical constraints in mind, and you should indicate which methods you favor, and why.

18.8. Prepare a summary of nonmedical applications of registration and pose consistency.

Programming Assignments

18.9. Representing an object as a linear combination of models is often represented as abstraction because we can regard adjusting the coefficients as obtaining the same view of different models. Furthermore, we could get a parametric family of models by adding a basis element to the space. Explore these ideas by building a system for matching rectangular buildings where the width, height, and depth of the building are unknown parameters. You should extend the linear combinations idea to handle orthographic cameras; this involves constraining the coefficients to represent rotations.

C H A P T E R 19

Smooth Surfaces and Their Outlines

PROBLEMS

19.1. What is (in general) the shape of the silhouette of a sphere observed by a perspective camera?

Solution The silhouette of a sphere observed by a perspective camera is the intersection of the corresponding viewing cone with the image plane. By symmetry, this cone is circular and grazes the sphere along a circular occluding contour. The silhouette is therefore the intersection of a circular cone with a plane, i.e., a conic section. For most viewing situations this conic section is an ellipse. It is a circle when the image plane is perpendicular to the axis of the cone. It may also be a parabola or even a hyperbola branch for extreme viewing angles.

19.2. What is (in general) the shape of the silhouette of a sphere observed by an orthographic camera?

Solution Under orthographic projection, the viewing cone degenerates into a viewing cylinder. By symmetry this cylinder is circular. Since the image plane is perpendicular to the projection direction under orthographic projection, the silhouette of the sphere is the intersection of a circular cylinder with a plane perpendicular to its axis, i.e., a circle.

19.3. Prove that the curvature κ of a planar curve in a point P is the inverse of the radius of curvature r at this point.
Hint: Use the fact that $\tan u \approx u$ for small angles.

Solution As P' approaches P the direction of the line PP' approaches that of the tangent T, and δs is, to first order, equal to the distance between P and P', it follows that $PM \approx PP'/\tan\delta\theta \approx \delta s/\delta\theta$. Passing to the limit, we obtain that the curvature is the inverse of the radius of curvature.

19.4. Given a fixed coordinate system, let us identify points of $\mathbb{E}^3$ with their coordinate vectors and consider a parametric curve $\boldsymbol{x} : I \subset \mathbb{R} \to \mathbb{R}^3$ not necessarily parameterized by arc length. Show that its curvature is given by

$$\kappa = \frac{|\boldsymbol{x}' \times \boldsymbol{x}''|}{|\boldsymbol{x}'|^3}, \tag{19.1}$$

where $\boldsymbol{x}'$ and $\boldsymbol{x}''$ denote, respectively, the first and second derivatives of $\boldsymbol{x}$ with respect to the parameter t defining it.
Hint: Reparameterize $\boldsymbol{x}$ by its arc length and reflect the change of parameters in the differentiation.

Solution We can write

$$\boldsymbol{x}' = \frac{d}{dt}\boldsymbol{x} = \frac{ds}{dt}\frac{d}{ds}\boldsymbol{x} = \frac{ds}{dt}\boldsymbol{t},$$

and

$$\boldsymbol{x}'' = \frac{d}{dt}\boldsymbol{x}' = \frac{d^2s}{dt^2}\boldsymbol{t} + (\frac{ds}{dt})^2\frac{d}{ds}\boldsymbol{t} = \frac{d^2s}{dt^2}\boldsymbol{t} + \kappa(\frac{ds}{dt})^2\boldsymbol{n}.$$

It follows that

$$\boldsymbol{x}' \times \boldsymbol{x}'' = \kappa(\frac{ds}{dt})^3\boldsymbol{b},$$

and since $\boldsymbol{t}$ and $\boldsymbol{b}$ have unit norm, we have indeed

$$\kappa = \frac{|\boldsymbol{x}' \times \boldsymbol{x}''|}{|\boldsymbol{x}'|^3}.$$

19.5. Prove that, unless the normal curvature is constant over all possible directions, the principal directions are orthogonal to each other.

Solution According to Ex. 19.6 below, the second fundamental form is symmetric. If follows that the tangent plane admits an orthonormal basis formed by the eigenvectors of the associated linear map $d\boldsymbol{N}$, and that the corresponding eigenvalues are real (this is a general property of symmetric operators). Unless they are equal, the orthonormal basis is essentially unique (except for swapping the two eigenvectors or changing their orientation), and the two eigenvalues are the maximum and minimum values of the second fundamental form (this is another general property of quadratic forms, see chapter 3 for a proof that the maximum value of a quadratic form is the maximum eigenvalue of the corresponding linear map). It follows that the principal curvatures are the two eigenvalues, and the principal directions are the corresponding eigenvectors, that are uniquely defined (in the sense used above) and orthogonal to each other unless the eigenvalues are equal, in which case the normal curvature is constant.

19.6. Prove that the second fundamental form is bilinear and symmetric.

Solution The bilinearity of the second fundamental form follows immediately from the fact that the differential of the Gauss map is linear. We remain quite informal in our proof of its symmetry. Given two directions $\boldsymbol{u}$ and $\boldsymbol{v}$ in the tangent plane of a surface S at some point P_0, we pick a parameterization $P : U \times V \subset \mathbb{R}^2 \rightarrow W \subset S$ of S in some neighborhood W of P_0 such that $P(0,0) = P_0$, and the tangents to the two surface curves α and β respectively defined by $P(u,0)$ for $u \in I$ and $P(0,v)$ for $v \in J$ are respectively $\boldsymbol{u}$ and $\boldsymbol{v}$. We assume that this parameterization is differentiable as many times as desired and abstain from justifying its existence. We omit the parameters from now on and assume that all functions are evaluated in $(0,0)$. We use subscripts to denote partial derivatives, e.g., P_{uv} denotes the second partial derivative of P with respect to u and v. The partial derivatives P_u and P_v lie in the tangent plane at any point in W. Differentiating $\boldsymbol{N} \cdot P_u =$ with respect to v yields

$$\boldsymbol{N}_v \cdot P_u + \boldsymbol{N} \cdot P_{uv} = 0.$$

Likewise, we have

$$\boldsymbol{N}_u \cdot P_v + \boldsymbol{N} \cdot P_{vu} = 0.$$

But since the cross derivatives are equal, we have

$$\boldsymbol{N}_u \cdot P_v = \boldsymbol{N}_v \cdot P_u,$$

or equivalently

$$\boldsymbol{v} \cdot d\boldsymbol{N}\boldsymbol{u} = \boldsymbol{u} \cdot d\boldsymbol{N}\boldsymbol{v},$$

which shows that the second fundamental form is indeed symmetric.

19.7. Let us denote by α the angle between the plane Π and the tangent to a curve Γ and by β the angle between the normal to Π and the binormal to Γ, and by κ the curvature at some point on Γ. Prove that if κ_a denotes the apparent curvature of the image of Γ at the corresponding point, then

$$\kappa_a = \kappa \frac{\cos\beta}{\cos^3\alpha}.$$

(Note: This result can be found in Koenderink, 1990, p. 191.)
Hint: Write the coordinates of the vectors $\boldsymbol{t}$, $\boldsymbol{n}$, and $\boldsymbol{b}$ in a coordinate system whose z-axis is orthogonal to the image plane, and use Eq. (19.6) to compute κ_a.

Solution Let us consider a particular point P_0 on the curve and pick the coordinate system $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ so that $(\boldsymbol{i}, \boldsymbol{j})$ is a basis for the image plane with $\boldsymbol{i}$ along the projection of the tangent $\boldsymbol{t}$ to the curve in P_0. Given this coordinate system, let us now identify curve points with their coordinate vectors, and parameterize Γ by its arc length s in the neighborhood of P_0. Let us denote by $\boldsymbol{x} : I \subset \mathbb{R} \to \mathbb{R}^3$ this parametric curve and by $\boldsymbol{y} : I \subset \mathbb{R} \to \mathbb{R}^3$ its orthographic projection. We omit the parameter s from now on and write

$$\begin{cases} \boldsymbol{y} = (\boldsymbol{x} \cdot \boldsymbol{i})\boldsymbol{i} + (\boldsymbol{x} \cdot \boldsymbol{j})\boldsymbol{j}, \\ \boldsymbol{y}' = (\boldsymbol{t} \cdot \boldsymbol{i})\boldsymbol{i} + (\boldsymbol{t} \cdot \boldsymbol{j})\boldsymbol{j}, \\ \boldsymbol{y}'' = \kappa[(\boldsymbol{n} \cdot \boldsymbol{i})\boldsymbol{i} + (\boldsymbol{n} \cdot \boldsymbol{j})\boldsymbol{j}]. \end{cases}$$

Recall that the curvature κ_a of $\boldsymbol{y}$ is, according to Ex. 19.4,

$$\kappa_a = \frac{|\boldsymbol{y}' \times \boldsymbol{y}''|}{|\boldsymbol{y}'|^3}$$

By construction, we have $\boldsymbol{t} = (\cos\alpha, 0, \sin\alpha)$. Thus $\boldsymbol{y}' = \cos\alpha\boldsymbol{i}$. Now, if $\boldsymbol{n} = (a, b, c)$, we have $\boldsymbol{b} = \boldsymbol{t} \times \boldsymbol{n} = (-b\sin\alpha, a\sin\alpha - c\cos\alpha, b\cos\alpha)$, and since the angle between the projection direction and $\boldsymbol{b}$ is β, we have $b = \cos\beta/\cos\alpha$. It follows that

$$\boldsymbol{y}' \times \boldsymbol{y}'' = \kappa[(\boldsymbol{t} \cdot \boldsymbol{i})(\boldsymbol{n} \cdot \boldsymbol{j}) - (\boldsymbol{t} \cdot \boldsymbol{j})(\boldsymbol{n} \cdot \boldsymbol{i})]\boldsymbol{k} = \kappa b \cos\alpha \boldsymbol{k} = \kappa \cos\beta \boldsymbol{k}.$$

Putting it all together we finally obtain

$$\kappa_a = \frac{|\cos\beta|}{|\cos^3\alpha|},$$

but α can always be taken positive (just pick the appropriate orientation for $\boldsymbol{i}$), and β can also be taken positive by choosing the orientation of Γ appropriately. The result follows.

19.8. Let $\kappa_{\boldsymbol{u}}$ and $\kappa_{\boldsymbol{v}}$ denote the normal curvatures in conjugated directions $\boldsymbol{u}$ and $\boldsymbol{v}$ at a point P, and let K denote the Gaussian curvature; prove that

$$K \sin^2\theta = \kappa_{\boldsymbol{u}} \kappa_{\boldsymbol{v}},$$

where θ is the angle between the $\boldsymbol{u}$ and $\boldsymbol{v}$.

Hint: Relate the expressions obtained for the second fundamental form in the bases of the tangent plane respectively formed by the conjugated directions and the principal directions.

Solution Let us assume that $\boldsymbol{u}$ and $\boldsymbol{v}$ are unit vectors and write them in the basis of the tangent plane formed by the (unit) principal directions as $\boldsymbol{u} = u_1\boldsymbol{e}_1 + u_2\boldsymbol{e}_2$ and $\boldsymbol{v} = v_1\boldsymbol{e}_1 + v_2\boldsymbol{e}_2$. We have

$$\begin{cases} \mathrm{II}(\boldsymbol{u}, \boldsymbol{u}) = \kappa_{\boldsymbol{u}} = \kappa_1 u_1^2 + \kappa_2 u_2^2, \\ \mathrm{II}(\boldsymbol{v}, \boldsymbol{v}) = \kappa_{\boldsymbol{v}} = \kappa_1 v_1^2 + \kappa_2 v_2^2, \\ \mathrm{II}(\boldsymbol{u}, \boldsymbol{v}) = 0 = \kappa_1 u_1 v_1 + \kappa_2 u_2 v_2. \end{cases}$$

According to the third equation, we have $\kappa_2 = -\kappa_1 u_1 v_1 / u_2 v_2$. Substituting this value in the first equation yields

$$\kappa_{\boldsymbol{u}} = \kappa_1 \frac{u_1}{v_2}(u_1 v_2 - u_2 v_1) = \kappa_1 \frac{u_1}{v_2} \sin\theta$$

since $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ form an orthonormal basis of the tangent plane and $\boldsymbol{u}$ has unit norm. A similar line of reasoning shows that

$$\kappa_{\boldsymbol{v}} = \kappa_2 \frac{v_2}{u_1}(u_1 v_2 - u_2 v_1) = \kappa_2 \frac{v_2}{u_1} \sin\theta,$$

and we finally obtain $\kappa_{\boldsymbol{u}}\kappa_{\boldsymbol{v}} = \kappa_1\kappa_2 \sin^2\theta = K \sin^2\theta$.

19.9. Show that the occluding is a smooth curve that does not intersect itself.
Hint: Use the Gauss map.

Solution Suppose that the occluding contour has a tangent discontinuity or a self intersection at some point P. In either case, two branches of the occluding contour meet in P with distinct tangents $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, and the viewing direction $\boldsymbol{v}$ must be conjugated with both directions. But $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ form a basis of the tangent plane, and by linearity $\boldsymbol{v}$ must be conjugated with all directions of the tangent plane. This cannot happen until the point is planar, in which case $d\boldsymbol{N}$ is zero, or $\boldsymbol{v}$ is an asymptotic direction at a parabolic point, in which case $d\boldsymbol{N}\boldsymbol{v} = \boldsymbol{0}$. Neither situation occurs for generic observed from generic viewpoints.

19.10. Show that the apparent curvature of any surface curve with tangent $\boldsymbol{t}$ is

$$\kappa_a = \frac{\kappa_{\boldsymbol{t}}}{\cos^2\alpha},$$

where α is the angle between the image plane and $\boldsymbol{t}$.
Hint: Write the coordinates of the vectors $\boldsymbol{t}$, $\boldsymbol{n}$, and $\boldsymbol{b}$ in a coordinate system whose z axis is orthogonal to the image plane, and use Eq. (19.2) and Meusnier's theorem.

Solution Let us denote by Γ the surface curve and by γ its projection. We assume of course that the point P that we are considering lies on the occluding contour (even though the curve under consideration may not be the occluding contour). Since $\kappa_{\boldsymbol{t}}$ is a signed quantity, it will be necessary to give κ_a a meaningful sign to establish the desired result. Let us first show that, whatever that meaning may be, we have indeed

$$|\kappa_a| = \frac{|\kappa_{\boldsymbol{t}}|}{\cos^2\alpha}.$$

We follow the notation of Ex. 19.7 and use the same coordinate system. Since P is on the occluding contour, the surface normal $\boldsymbol{N}$ in P is also the normal to γ, and we must have $\boldsymbol{N} = \mp\boldsymbol{j}$. Let ϕ denote the angle between $\boldsymbol{N}$ and the principal normal $\boldsymbol{n}$ to Γ. We must therefore have $b = |\cos\phi| = cos\beta/\cos\alpha$ (since we have chosen our coordinate system so $\cos\alpha \geq 0$ and $\cos\beta \geq 0$), and it follows, according to Meusnier's theorem and Ex. 19.7, that

$$|\kappa_a| = \kappa\frac{\cos\beta}{\cos^3\alpha} = |\kappa_{\boldsymbol{t}}\cos\phi|\frac{\cos\beta}{\cos^3\alpha} = \frac{|\kappa_{\boldsymbol{t}}|}{\cos^2\alpha}.$$

Let us now turn to giving a meaningful sign to κ_a and determining this sign. By convention, we take κ_a positive when the principal normal $\boldsymbol{n}'$ to γ is equal to $-\boldsymbol{N}$, and negative when $\boldsymbol{n}' = \boldsymbol{N}$.
It is easy to show that with our choice of coordinate system, we always have $\boldsymbol{n}' = \boldsymbol{j}$: Briefly, let us reparameterize $\boldsymbol{y}$ by its arc length s', noting that, because of the foreshortening induced by the projection, $ds' = ds\cos\alpha$. Using a line or reasoning similar to Ex. 19.7 but differentiating $\boldsymbol{y}$ with respect to s', it is easy to show that the cross product of the tangent $\boldsymbol{t}' = \boldsymbol{i}$ and (principal) normal $\boldsymbol{n}'$ to γ verify

$$\boldsymbol{t}' \times (\kappa'\boldsymbol{n}') = \kappa\frac{\cos\beta}{\cos^3\alpha}\boldsymbol{k},$$

where κ' is the (nonnegative) curvature of γ. In particular, the vectors $\boldsymbol{t}' = \boldsymbol{i}$, $\boldsymbol{n}' = \mp\boldsymbol{j}$, and $\boldsymbol{k}$ must form a right-handed coordinate system, which implies $\boldsymbol{n}' = \boldsymbol{j}$. Therefore we must take $\kappa_a > 0$ when $\boldsymbol{N} = -\boldsymbol{j}$, and $\kappa_a < 0$ when $\boldsymbol{N} = \boldsymbol{j}$. Suppose that $\boldsymbol{N} = -\boldsymbol{j}$, then $\cos\phi = \boldsymbol{N}\cdot\boldsymbol{n} = -\boldsymbol{j}\cdot\boldsymbol{n} = -b$ must be negative, and by Meusnier's theorem, $\kappa_{\boldsymbol{t}}$ must be positive. By the same token, when $\boldsymbol{N} = \boldsymbol{j}$, $\cos\phi$ must be positive and $\kappa_{\boldsymbol{t}}$ must be negative. It follows that we can indeed take

$$\kappa_a = \frac{\kappa_{\boldsymbol{t}}}{\cos^2\alpha}.$$

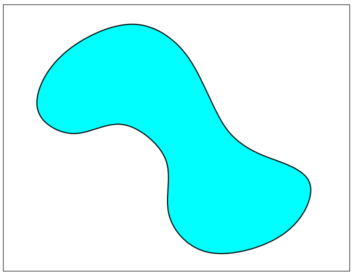
Note that when Γ is the occluding contour, the convention we have chosen for the sign of the apparent curvature yields the expected result: κ_a is positive when the contour point is convex (i.e., its principal normal is (locally) inside the region bounded by the image contour), and κ_a is negative when the point is concave.

C H A P T E R 20

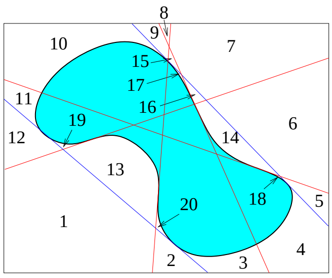
Aspect Graphs

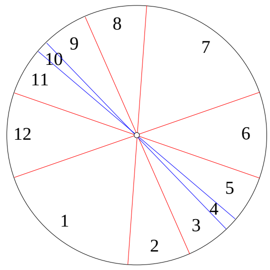
PROBLEMS

20.1. Draw the orthographic and spherical perspective aspect graphs of the transparent Flatland object below along with the corresponding aspects.

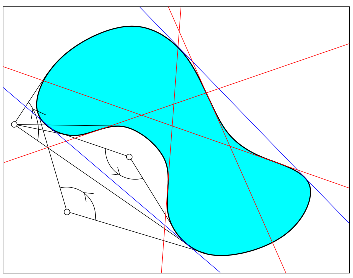


Solution The visual events for the transparent object, along with the various cells of the perspective and orthographic aspect graph, are shown below.



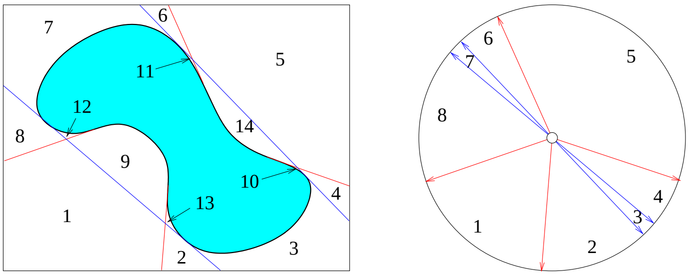


Note that cells of the perspective aspect graph created by the intersection of visual event rays outside of the box are not shown. Three of the perspective aspects are shown below. Note the change in the order of the contour points between aspects 1 and 13, and the addition of two contour points as one goes from aspect 1 to aspect 12.

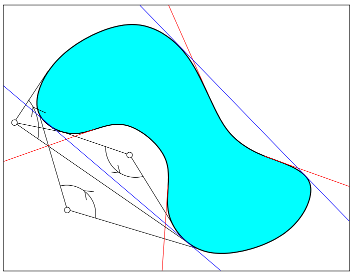


20.2. Draw the orthographic and spherical perspective aspect graphs of the opaque object along with the corresponding aspects.

Solution The visual events for the opaque object, along with the various cells of the perspective and orthographic aspect graph, are shown below.

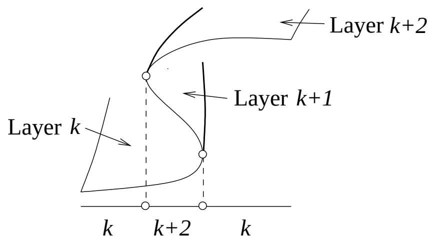


Note that cells of the perspective aspect graph created by the intersection of visual event rays outside of the box are not shown. Three of the perspective aspects are shown below. Note the change in the order of the contour points between aspects 1 and 13, and the addition of a single contour point as one goes from aspect 1 to aspect 12.



20.3. Is it possible for an object with a single parabolic curve (such as a banana) to have no cusp of Gauss at all? Why (or why not)?

Solution No it is not possible. To see why, consider a nonconvex compact solid. It is easy to see that the surface bounding this solid must have at least one convex point, so there must exist a parabolic curve separating this point from the nonconvex part of the surface. On the Gaussian sphere, as one crosses the image of this curve (the fold) at some point, the multiplicity of the sphere covering goes from k to $k+2$ with $k \geq 1$, or from k to $k-2$ with $k \geq 3$. Let us choose the direction of traversal so we go from k to $k+2$. Locally, the fold separates layers $k+1$ and $k+2$ of the sphere covering (see diagram below).



If there is no cusp, the fold is a smooth closed curve that forms (globally) the boundary between layers $k+1$ and $k+2$ (the change in multiplicity cannot change along the curve). But layer $k+1$ must be connected to layer k by another fold curve. Thus either the surface of a nonconvex compact solid admits cusps of Gauss, or it has at least two distinct parabolic curves.

20.4. Use an equation-counting argument to justify the fact that contact of order six or greater between lines and surfaces does not occur for generic surfaces. (Hint: Count the parameters that define contact.)

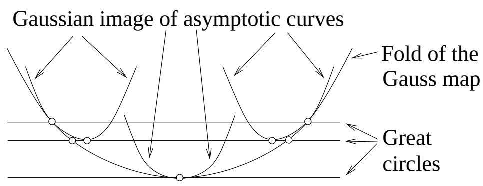
Solution A line has contact of order n with a surface when all derivatives of order less than or equal to $n-1$ of the surface are zero in the direction of the line. Ordinary tangents have order-two contact with the surface, and there is a three-parameter family of those (all tangent lines in the tangent planes of all surface points); asymptotic tangents have order-three contact and there is a two-parameter family of those (the two asymptotic tangents at each saddle-shaped point); order-four contact occurs for the asymptotic tangents along flecnodal and parabolic curves; there are a finite number of order-five tangents at isolated points of the surface (including gutterpoints and cusps of Gauss); and finally there is in general no order-six tangent.

20.5. We saw that the asymptotic curve and its spherical image have perpendicular tangents. Lines of curvature are the integral curves of the field of principal directions. Show that these curves and their Gaussian image have parallel tangents.

Solution Let us consider a line of curvature Γ parameterized by its arc length. Its unit tangent at some point P is by definition a principal direction $\boldsymbol{e}_i$ in P. Let κ_i denote the corresponding principal curvature. Since principal directions are the eigenvectors of the differential of the Gauss map, the derivative of the unit surface normal along Γ is $d\boldsymbol{N}\boldsymbol{e}_i = \kappa_i \boldsymbol{e}_i$. This is also the tangent to the Gaussian image of the principal curve, and the result follows.

20.6. Use the fact that the Gaussian image of a parabolic curve is the envelope of the asymptotic curves intersecting it to give an alternate proof that a pair of cusps is created (or destroyed) in a lip or beak-to-beak event.

Solution Lip and beak-to-beak events occur when the Gaussian image of the occluding contour becomes tangent to the fold associated with a parabolic point. Let us assume that the fold is convex at this point (a similar reasoning applies when the fold is concave, but the situation becomes more complicated at inflections). There exists some neighborhood of the tangency point such that any great circle intersecting the fold in this neighborhood will intersect it exactly twice. As illustrated by the diagram below, two of the asymptotic curve branches tangent to the fold at the intersections admit a great circle bitangent to them.



This great circle also intersects the fold exactly twice, and since it is tangent to the asymptotic curves, it is orthogonal to the corresponding asymptotic direction. In other words, the viewing direction is an asymptotic direction at the corresponding points of the occluding contour, yielding two cusps of the image contour.

20.7. Lip and beak-to-beak events of implicit surfaces. It can be shown (Pae and Ponce, 2001) that the parabolic curves of a surface defined implicitly as the zero set of some density function $F(x, y, z) = 0$ are characterized by this equation and $P(x, y, z) = 0$, where $P \stackrel{\text{def}}{=} \nabla F^T \mathcal{A} \nabla F = 0$, ∇F is the gradient of F, and $\mathcal{A}$ is the symmetric matrix

$$\mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix} F_{yy}F_{zz} - F_{yz}^2 & F_{xz}F_{yz} - F_{zz}F_{xy} & F_{xy}F_{yz} - F_{yy}F_{xz} \\ F_{xz}F_{yz} - F_{zz}F_{xy} & F_{zz}F_{xx} - F_{xz}^2 & F_{xy}F_{xz} - F_{xx}F_{yz} \\ F_{xy}F_{yz} - F_{yy}F_{xz} & F_{xy}F_{xz} - F_{xx}F_{yz} & F_{xx}F_{yy} - F_{xy}^2 \end{pmatrix}.$$

It can also be shown that the asymptotic direction at a parabolic point is $\mathcal{A}\nabla F$.

(a) Show that $\mathcal{A}\mathcal{H} = \text{Det}(\mathcal{H})\text{Id}$, where $\mathcal{H}$ denotes the Hessian of F.

(b) Show that cusps of Gauss are parabolic points that satisfy the equation $\nabla P^T \mathcal{A} \nabla F = 0$. Hint: Use the fact that the asymptotic direction at a cusp of Gauss is tangent to the parabolic curve, and that the vector ∇F is normal to the tangent plane of the surface defined by $F = 0$.

(c) Sketch an algorithm for tracing the lip and beak-to-beak events of an implicit surface.

Solution

(a) Note that the Hessian can be written as

$$\mathcal{H} = \begin{pmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 & \boldsymbol{h}_3 \end{pmatrix} \text{ where } \boldsymbol{h}_1 = \begin{pmatrix} F_{xx} \\ F_{xy} \\ F_{xz} \end{pmatrix},\ \boldsymbol{h}_2 = \begin{pmatrix} F_{xy} \\ F_{yy} \\ F_{yz} \end{pmatrix},\ \boldsymbol{h}_3 = \begin{pmatrix} F_{xz} \\ F_{yz} \\ F_{zz} \end{pmatrix}.$$

But, if $\boldsymbol{a}^i$ denotes the i^{th} coordinate of $\boldsymbol{a}$, we can rewrite $\mathcal{A}$ as

$$\mathcal{A} = \begin{pmatrix} (\boldsymbol{h}_2 \times \boldsymbol{h}_3)^1 & (\boldsymbol{h}_2 \times \boldsymbol{h}_3)^2 & (\boldsymbol{h}_2 \times \boldsymbol{h}_3)^3 \\ (\boldsymbol{h}_3 \times \boldsymbol{h}_1)^1 & (\boldsymbol{h}_3 \times \boldsymbol{h}_1)^2 & (\boldsymbol{h}_3 \times \boldsymbol{h}_1)^3 \\ (\boldsymbol{h}_1 \times \boldsymbol{h}_2)^1 & (\boldsymbol{h}_1 \times \boldsymbol{h}_2)^2 & (\boldsymbol{h}_1 \times \boldsymbol{h}_2)^3 \end{pmatrix} = \begin{pmatrix} (\boldsymbol{h}_2 \times \boldsymbol{h}_3)^T \\ (\boldsymbol{h}_3 \times \boldsymbol{h}_1)^T \\ (\boldsymbol{h}_1 \times \boldsymbol{h}_2)^T \end{pmatrix},$$

and it follows that

$$\mathcal{A}\mathcal{H} = \begin{pmatrix} (\boldsymbol{h}_2 \times \boldsymbol{h}_3)^T \\ (\boldsymbol{h}_3 \times \boldsymbol{h}_1)^T \\ (\boldsymbol{h}_1 \times \boldsymbol{h}_2)^T \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 & \boldsymbol{h}_3 \end{pmatrix} = \text{Det}(\mathcal{H})\text{Id}$$

since the determinant of the matrix formed by three vectors is the dot product of the first vector with the cross product of the other two vectors.

(b) The parabolic curve can be thought of as the intersection of the two surfaces defined by $F(x, y, z) = 0$ and $P(x, y, z) = 0$. Its tangent lies in the intersection of the tangent planes of these two surfaces and is therefore orthogonal to the normals ∇F and ∇P. For a point to be a cusp of Gauss, this tangent must be along the asymptotic direction $\mathcal{A}\nabla F$, and we must therefore have $\nabla F^T \mathcal{A} \nabla F = 0$, which is automatically satisfied at a parabolic point, and $\nabla P^T \mathcal{A} \nabla F = 0$, which is the desired condition.

(c) To trace the lip and beak-to-beak events, simply use Algorithm 20.2 to trace the parabolic curve defined in $\mathbb{R}^3$ by the equations $F(x, y, z) = 0$ and $P(x, y, z) = 0$, computing for each point along this curve the vector $\mathcal{A}\nabla F$ as the corresponding asymptotic direction. The cusps of Gauss can be found by adding $\nabla P^T \mathcal{A} \nabla F = 0$ to these two equations and solving the corresponding system of three polynomial equations in three unknowns using homotopy continuation.

20.8. Swallowtail events of implicit surfaces. It can be shown that the asymptotic directions $\boldsymbol{a}$ at a hyperbolic point satisfy the two equations $\nabla F \cdot \boldsymbol{a} = 0$ and $\boldsymbol{a}^T \mathcal{H} \boldsymbol{a} = 0$, where $\mathcal{H}$ denotes the Hessian of F. These two equations simply indicate that the order of contact between a surface and its asymptotic tangents is at least equal to three. Asymptotic tangents along flecnodal curves have order-four contact with the surface, and this is characterized by a third equation, namely

$$\begin{pmatrix} \boldsymbol{a}^T \mathcal{H}_x \boldsymbol{a} \\ \boldsymbol{a}^T \mathcal{H}_y \boldsymbol{a} \\ \boldsymbol{a}^T \mathcal{H}_z \boldsymbol{a} \end{pmatrix} \cdot \boldsymbol{a} = 0.$$

Sketch an algorithm for tracing the swallowtail events of an implicit surface.

Solution To trace the swallowtail events, one can use Algorithm 20.2 to trace the curve defined in $\mathbb{R}^6$ by $F(x, y, z) = 0$, the three equations given above, and $|\boldsymbol{a}|^2 = 1$. There are of course six unknowns in that case, namely x, y, z, and the three coordinates of $\boldsymbol{a}$. The corresponding visual events are then given by the values of $\boldsymbol{a}$ along the curve. Alternatively, one can use a computer algebra system to eliminate $\boldsymbol{a}$ among the three equations involving it. This yields an equation $S(x, y, z) = 0$, and the flecnodal curve can be traced as the zero set of $F(x, y, z)$ and $S(x, y, z)$. The corresponding visual events are then found by computing, for each flecnodal point, the singular asymptotic directions as the solutions of the three equations given above. Since these equations are homogeneous, $\boldsymbol{a}$ can be computed from any two of them. The singular asymptotic direction is the common solution of two of the equation pairs.

20.9. Derive the equations characterizing the multilocal events of implicit surfaces. You can use the fact that, as mentioned in the previous exercise, the asymptotic directions $\boldsymbol{a}$ at a hyperbolic point satisfy the two equations $\nabla F \cdot \boldsymbol{a} = 0$ and $\boldsymbol{a}^T \mathcal{H} \boldsymbol{a} = 0$.

Solution Triple points are characterized by the following equations in the positions of the contact points $\boldsymbol{x}_i = (x_i, y_i, z_i)^T$ $(i = 1, 2, 3)$:

$$\begin{cases} F(\boldsymbol{x}_i) = 0, \quad i = 1, 2, 3, \\ (\boldsymbol{x}_1 - \boldsymbol{x}_2) \cdot \nabla F(x_i) = 0, \quad i = 1, 2, 3, \\ (\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_1) = \boldsymbol{0}. \end{cases}$$

Note that the vector equation involving the cross product is equivalent to two independent scalar equations, thus triple points correspond to curves defined in $\mathbb{R}^9$ by eight equations in nine unknowns.

Tangent crossings correspond to curves defined in $\mathbb{R}^6$ by the following five equations in the positions of the contact points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

$$\begin{cases} F(\boldsymbol{x}_i) = 0, \quad i = 1, 2, \\ (\boldsymbol{x}_1 - \boldsymbol{x}_2) \cdot \nabla F(x_i), \quad i = 1, 2, \\ (\nabla F(x_1) \times \nabla F(x_2)) \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_1) = 0. \end{cases}$$

Finally, cusps crossings correspond to curves defined in $\mathbb{R}^6$ by the following five equations in the positions of the contact points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

$$\begin{cases} F(\boldsymbol{x}_i) = 0, \quad i = 1, 2, \\ (\boldsymbol{x}_1 - \boldsymbol{x}_2) \cdot \nabla F(x_i) = 0, \quad i = 1, 2, \\ (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T \mathcal{H}(\boldsymbol{x}_1)(\boldsymbol{x}_2 - \boldsymbol{x}_1) = 0, \end{cases}$$

where the last equation simply expresses the fact that the viewing direction is an asymptotic direction of the surface in $\boldsymbol{x}_1$.

Programming Assignments

20.10. Write a program to explore multilocal visual events: Consider two spheres with different radii and assume orthographic projection. The program should allow you to change viewpoint interactively as well as explore the tangent crossings associated with the limiting bitangent developable.

20.11. Write a similar program to explore cusp points and their projections. You have to trace a plane curve.

CHAPTER 21

Range Data

PROBLEMS

21.1. Use Eq. (21.1) to show that a necessary and sufficient condition for the coordinate curves of a parameterized surface to be lines of curvature is that $f = F = 0$.

Solution The principal directions satisfy the differential equation (21.1) reproduced here for completeness:

$$0 = \begin{vmatrix} v'^2 & -u'v' & u'^2 \\ E & F & G \\ e & f & g \end{vmatrix} = (Fg - fG)v'^2 - u'v'(Ge - gE) + u'^2(Ef - eF).$$

The coordinate curves are lines of curvature when their tangents are along principal directions. Equivalently, the solutions of Eq. (21.1) must be $u' = 0$ and $v' = 0$, which is in turn equivalent to $Fg - fG = Ef - eF = 0$. Clearly, this condition is satisfied when $f = F = 0$. Conversely, when $Fg - fG = Ef - eF = 0$, either $f = F = 0$, or we can write $e = \lambda E$, $f = \lambda F$, and $g = \lambda G$ for some scalar $\lambda \neq 0$. In the latter case, the normal curvature in any direction $\boldsymbol{t}$ of the tangent plane is given by

$$\kappa_{\boldsymbol{t}} = \frac{\mathrm{II}(\boldsymbol{t}, \boldsymbol{t})}{\mathrm{I}(\boldsymbol{t}, \boldsymbol{t})} = \frac{eu'^2 + 2fu'v' + gv'^2}{Eu'^2 + 2Fu'v' + Gv'^2} = \lambda,$$

i.e., the normal curvature is independent of the direction in which it is measured (we say that such a point is an *umbilic*). In this case, the principal directions are of course ill defined. It follows that a necessary and sufficient condition for the coordinate curves of a parameterized surface to be lines of curvature is that $f = F = 0$.

21.2. Show that the lines of curvature of a surface of revolution are its meridians and parallels.

Solution Let us consider a surface of revolution parameterized by $\boldsymbol{x}(\theta, z) = (r(z)\cos\theta, r(z)\sin\theta, z)^T$, where $r(z_0)$ denotes the radius of the circle formed by the intersection of the surface with the plane $z = z_0$. We have $\boldsymbol{x}_\theta = (-r\sin\theta, r\cos\theta, 0)^T$ and $\boldsymbol{x}_z = (r'\cos\theta, r'\sin\theta, 1)^T$, thus $F = \boldsymbol{x}_\theta \cdot \boldsymbol{x}_z = 0$. Now, normalizing the cross product of $\boldsymbol{x}_\theta$ and $\boldsymbol{x}_z$ shows that $\boldsymbol{N} = (1/\sqrt{1 + r'^2})(\cos\theta, \sin\theta, -r')^T$. Finally, we have $\boldsymbol{x}_{\theta z} = (-r'\sin\theta, r'\cos\theta, 0)^T$, and it follows that $f = -\boldsymbol{N} \cdot \boldsymbol{x}_{\theta z} = 0$. According to the previous exercise, the lines of curvature of a surface of revolution are thus its coordinate curves—that is, its meridians and its parallels.

21.3. Step model: Compute $z_\sigma(x) = G_\sigma * z(x)$, where $z(x)$ is given by Eq. (21.2). Show that z''_σ is given by Eq. (21.3). Conclude that $\kappa''_\sigma / \kappa'_\sigma = -2\delta/h$ in the point x_σ where z''_σ and κ_σ vanish.

Solution Recall that the step model is defined by

$$z = \begin{cases} k_1 x + c & \text{when} \quad x < 0, \\ k_2 x + c + h & \text{when} \quad x > 0. \end{cases}$$

Convolving it with G_σ yields

$$z_\sigma = c + \frac{h}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x}\exp(-\frac{t^2}{2\sigma^2})dt + kx + \frac{\delta x}{\sigma\sqrt{2\pi}}\int_0^x \exp(-\frac{t^2}{2\sigma^2})dt + \frac{\delta\sigma}{\sqrt{2\pi}}\exp(-\frac{x^2}{2\sigma^2}),$$

and the first and second derivatives of z_σ are respectively

$$z'_\sigma = k + \frac{\delta}{\sigma\sqrt{2\pi}}\int_0^x \exp(-\frac{t^2}{2\sigma^2})dt + \frac{h}{\sigma\sqrt{2\pi}}\exp(-\frac{x^2}{2\sigma^2})$$

and

$$z''_\sigma = \frac{1}{\sigma\sqrt{2\pi}}(\delta - \frac{hx}{\sigma^2})\exp(-\frac{x^2}{2\sigma^2}).$$

The latter is indeed Eq. (21.3). Now, we have

$$\kappa'_\sigma = \frac{z'''_\sigma}{[1+z'^2_\sigma]^{3/2}} - 3\frac{z'_\sigma z''_\sigma}{[1+z'^2_\sigma]^3}.$$

Since $z''_\sigma(x_\sigma) = 0$, the second term in the expression above vanishes in x_σ. Now, the derivatives of the numerator and denominator of this term obviously also vanish in x_σ since all the terms making them up contain z''_σ as a factor. Likewise, the derivative of the denominator of the first term vanished in x_σ, and it follows that $\kappa''_\sigma/\kappa'_\sigma = z''''_\sigma/z'''_\sigma$ at this point. Now, we can write $z''_\sigma = a\exp(-x^2/2\sigma^2)$, with $a = (\delta - xh/\sigma^2)/\sigma\sqrt{2\pi}$. Therefore,

$$z'''_\sigma = (a' - \frac{ax}{\sigma^2})\exp(-\frac{x^2}{2\sigma^2})$$

and

$$z''''_\sigma = (a'' - \frac{a}{\sigma^2}(1-\frac{x^2}{\sigma^2}) - \frac{2a'x}{\sigma^2})\exp(-\frac{x^2}{2\sigma^2}).$$

Now, a'' is identically zero, and a is zero in x_σ. It follows that $\kappa''_\sigma/\kappa'_\sigma = -2x_\sigma/\sigma^2 = -2\delta/h$ at this point.

21.4. Roof model: Show that κ_σ is given by Eq. (21.4).

Solution Plugging the value $h = 0$ in the expressions for z'_σ and z''_σ derived in the previous exercise shows immediately that

$$\kappa_\sigma = \frac{1}{\sigma\sqrt{2\pi}}\frac{\delta\exp(-x^2/2\sigma^2)}{\left[1+\left(k+\frac{\delta}{\sigma\sqrt{2\pi}}\int_0^x \exp(-t^2/2\sigma^2)dt\right)^2\right]^{3/2}},$$

and using the change of variable $u = t/\sigma$ in the integral finally shows that

$$\kappa_\sigma = \frac{1}{\sigma\sqrt{2\pi}}\frac{\delta\exp(-\frac{x^2}{2\sigma^2})}{\left[1+\left(k+\frac{\delta}{\sqrt{2\pi}}\int_0^{x/\sigma}\exp(-\frac{u^2}{2})du\right)^2\right]^{3/2}}.$$

21.5. Show that the quaternion $\mathsf{q} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\boldsymbol{u}$ represents the rotation $\mathcal{R}$ of angle θ about the unit vector $\boldsymbol{u}$ in the sense of Eq. (21.5).
Hint: Use the Rodrigues formula derived in the exercises of chapter 3.

Solution Let us consider some vector $\boldsymbol{\alpha}$ in $\mathbb{R}^3$, define $\boldsymbol{\beta} = \mathsf{q}\boldsymbol{\alpha}\bar{\mathsf{q}}$, and show that $\boldsymbol{\beta}$ is the vector of $\mathbb{R}^3$ obtained by rotating $\boldsymbol{\alpha}$ about the vector $\boldsymbol{u}$ by an angle θ. Recall that the quaternion product is defined by

$$(a + \boldsymbol{\alpha})(b + \boldsymbol{\beta}) \stackrel{\text{def}}{=} (ab - \boldsymbol{\alpha}\cdot\boldsymbol{\beta}) + (a\boldsymbol{\beta} + b\boldsymbol{\alpha} + \boldsymbol{\alpha}\times\boldsymbol{\beta}).$$

Thus,

$$\begin{aligned}
\boldsymbol{\beta} &= [\cos\tfrac{\theta}{2} + \sin\tfrac{\theta}{2}\boldsymbol{u}]\boldsymbol{\alpha}[\cos\tfrac{\theta}{2} - \sin\tfrac{\theta}{2}\boldsymbol{u}] \\
&= [\cos\tfrac{\theta}{2} + \sin\tfrac{\theta}{2}\boldsymbol{u}][\sin\tfrac{\theta}{2}(\boldsymbol{\alpha}\cdot\boldsymbol{u}) + \cos\tfrac{\theta}{2}\boldsymbol{\alpha} - \sin\tfrac{\theta}{2}\boldsymbol{\alpha}\times\boldsymbol{u}] \\
&= \cos^2\tfrac{\theta}{2}\boldsymbol{\alpha} + \sin^2\tfrac{\theta}{2}(\boldsymbol{u}\cdot\boldsymbol{\alpha})\boldsymbol{u} - 2\sin\tfrac{\theta}{2}\cos\tfrac{\theta}{2}(\boldsymbol{\alpha}\times\boldsymbol{u}) - \sin^2\tfrac{\theta}{2}\boldsymbol{u}\times(\boldsymbol{\alpha}\times\boldsymbol{u}) \\
&= \cos^2\tfrac{\theta}{2}\boldsymbol{\alpha} + \sin^2\tfrac{\theta}{2}(\boldsymbol{u}\cdot\boldsymbol{\alpha})\boldsymbol{u} - 2\sin\tfrac{\theta}{2}\cos\tfrac{\theta}{2}(\boldsymbol{\alpha}\times\boldsymbol{u}) + \sin^2\tfrac{\theta}{2}[\boldsymbol{u}_\times]^2\boldsymbol{\alpha}.
\end{aligned}$$

But remember from chapter 13 that $[\boldsymbol{u}_\times]^2 = \boldsymbol{u}\boldsymbol{u}^T - |\boldsymbol{u}|^2\mathrm{Id}$, so

$$\begin{aligned}
\boldsymbol{\beta} &= [\cos^2\tfrac{\theta}{2} - \sin^2\tfrac{\theta}{2}]\boldsymbol{\alpha} - 2\sin\tfrac{\theta}{2}\cos\tfrac{\theta}{2}(\boldsymbol{\alpha}\times\boldsymbol{u}) + 2\sin^2\tfrac{\theta}{2}(\boldsymbol{u}\cdot\boldsymbol{\alpha})\boldsymbol{u} \\
&= \cos\theta\boldsymbol{\alpha} + \sin\theta(\boldsymbol{u}\times\boldsymbol{\alpha}) + (1-\cos\theta)(\boldsymbol{u}\cdot\boldsymbol{\alpha})\boldsymbol{u},
\end{aligned}$$

which is indeed the Rodrigues formula for a rotation of angle θ about the unit vector $\boldsymbol{u}$.

21.6. Show that the rotation matrix $\mathcal{R}$ associated with a given unit quaternion $\mathsf{q} = a+\boldsymbol{\alpha}$ with $\boldsymbol{\alpha} = (b, c, d)^T$ is given by Eq. (21.6).

Solution First, note that any *unit* quaternion can be written as $\mathsf{q} = a + \boldsymbol{\alpha}$ where $a = \cos\frac{\theta}{2}$ and $\boldsymbol{\alpha} = \sin\frac{\theta}{2}\boldsymbol{u}^T$ for some angle θ and unit vector $\boldsymbol{u}$ (this is because $a^2 + |\boldsymbol{\alpha}|^2 = 1$). Now, to derive Eq. (21.6), i.e.,

$$\mathcal{R} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix},$$

all we need to do is combine the result established in the previous exercise with that obtained in Ex. 3.7, that states that if $\boldsymbol{u} = (u, v, w)^T$, then

$$\mathcal{R} = \begin{pmatrix} u^2(1-\cos\theta)+\cos\theta & uv(1-\cos\theta)-w\sin\theta & uw(1-\cos\theta)+v\sin\theta \\ uv(1-\cos\theta)+w\sin\theta & v^2(1-\cos\theta)+\cos\theta & vw(1-\cos\theta)-u\sin\theta \\ uw(1-\cos\theta)-v\sin\theta & vw(1-\cos\theta)+u\sin\theta & w^2(1-\cos\theta)+\cos\theta \end{pmatrix}.$$

Showing that the entries of both matrices are the same is a (slightly tedious) exercise in algebra and trigonometry.
Let us first show that the two top left entries are the same. Note that $|\boldsymbol{\alpha}|^2 = b^2 + c^2 + d^2 = \sin^2\frac{\theta}{2}|\boldsymbol{u}|^2 = \sin^2\frac{\theta}{2}$; it follows that

$$a^2+b^2-c^2-d^2 = a^2+2b^2-\sin^2\frac{\theta}{2} = \cos^2\frac{\theta}{2}-\sin^2\frac{\theta}{2}+2\sin^2\frac{\theta}{2}u^2 = \cos\theta+u^2(1-\cos\theta).$$

The exact same type of resoning applies to the other diagonal entries.
Let us not consider the entries corresponding to the first row and the second column of the two matrices. We have

$$2(bc-ad) = 2uv\sin^2\frac{\theta}{2} - 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}w = uv(1-\cos\theta) - w\sin\theta,$$

so the two entries do indeed coincide. The exact same type of resoning applies to the other non-diagonal entries.

21.7. Show that the matrix $\mathcal{A}_i$ constructed in Section 21.3.2 is equal to

$$\mathcal{A}_i = \begin{pmatrix} 0 & \boldsymbol{y}_i^T - \boldsymbol{y}_i'^T \\ \boldsymbol{y}_i' - \boldsymbol{y}_i & [\boldsymbol{y}_i + \boldsymbol{y}_i']_\times \end{pmatrix}.$$

Solution When $\mathsf{q} = a + \boldsymbol{\alpha}$ is a unit quaternion, we can write

$$\begin{aligned} \boldsymbol{y}_i'\mathsf{q} - \mathsf{q}\boldsymbol{y}_i &= (-\boldsymbol{y}_i' \cdot \boldsymbol{\alpha} + a\boldsymbol{y}_i' + \boldsymbol{y}_i' \times \boldsymbol{\alpha}) - (-\boldsymbol{y}_i \cdot \boldsymbol{\alpha} + a\boldsymbol{y}_i + \boldsymbol{\alpha} \times \boldsymbol{y}_i) \\ &= (\boldsymbol{y}_i - \boldsymbol{y}_i') \cdot \boldsymbol{\alpha} + a(\boldsymbol{y}_i' - \boldsymbol{y}_i) + (\boldsymbol{y}_i + \boldsymbol{y}_i') \times \boldsymbol{\alpha} \\ &= \begin{pmatrix} 0 & \boldsymbol{y}_i^T - \boldsymbol{y}_i'^T \\ \boldsymbol{y}_i' - \boldsymbol{y}_i & [\boldsymbol{y}_i + \boldsymbol{y}_i']_\times \end{pmatrix} \begin{pmatrix} a \\ \boldsymbol{\alpha} \end{pmatrix} = \mathcal{A}_i \mathsf{q}, \end{aligned}$$

where q has been identified with the 4-vector whose first coordinate is a and the remaining coordinates are those of $\boldsymbol{\alpha}$. In particular, we have

$$E = \sum_{i=1}^{n} |\boldsymbol{y}_i'\mathsf{q} - \mathsf{q}\boldsymbol{y}_i|^2 = \sum_{i=1}^{n} |\mathcal{A}_i\mathsf{q}|^2 = \mathsf{q}^T \mathcal{B}\mathsf{q}, \text{ where } \mathcal{B} = \sum_{i=1}^{n} \mathcal{A}_i^T \mathcal{A}_i.$$

21.8. As mentioned earlier, the ICP method can be extended to various types of geometric models. We consider here the case of polyhedral models and piecewise parametric patches.

(a) Sketch a method for computing the point Q in a polygon that is closest to some point P.

(b) Sketch a method for computing the point Q in the parametric patch $\boldsymbol{x} : I \times J \to \mathbb{R}^3$ that is closest to some point P. Hint: Use Newton iterations.

Solution

(a) Let A denote the polygon. Construct the orthogonal projection Q of P into the plane that contains A. Test whether Q lies inside A. When A is convex, this can be done by testing whether Q is on the "right" side of all the edges of A. When A is not convex, one can accumulate the angles between Q and the successive vertices of A. The point will be inside the polygon when these angles add to 2π, on the boundary when they add to π, and outside when they add to 0. Both methods take linear time. If Q is inside A, it is the closest point to P. If it is outside, the closest point must either lie in the interior of one of the edges (this is checked by projecting P onto the line supporting each edge) or be one of the vertices, and it can be found in linear time as well.

(b) Just as in the polygonal case, the shortest distance is reached either in the interior of the patch or on its boundary. We only detail the case where the closest point lies inside the patch, since the case where it lies in the interior of a boundare curve is similar, and the case where it is a vertex is straightforward. As suggested, starting from some point $\boldsymbol{x}(u, v)$ inside the patch, we can use Newton iterations to find the closest point, which is the orthogonal projection of P onto the patch. Thus we seek a zero of the (vector) function $\boldsymbol{f}(u, v) = \boldsymbol{N}(u, v) \times (\boldsymbol{x}(u, v) - \boldsymbol{P})$, where $\boldsymbol{P}$ denotes the coordinate vector of the point P. The Jacobian $\mathcal{J}$ of $\boldsymbol{f}$ is easily computed as a function of first- and second-order derivatives of $\boldsymbol{x}$. The increment $(\delta u, \delta v)$ is then computed as usual by solving

$$\mathcal{J}\begin{pmatrix}\delta u\\ \delta v\end{pmatrix} = -\boldsymbol{f}.$$

Note that this is an overconstrained system of equations ($\boldsymbol{f}$ has three components). It can be solved by discarding one of the redundant equations or by using the pseudoinverse of $\mathcal{J}$ to solve for δu and δv.

21.9. Develop a linear least-squares method for fitting a quadric surface to a set of points under the constraint that the quadratic form has unit Frobenius form.

Solution The equation of a quadric surface can be written as

$$\begin{aligned}&a_{200}x^2 + 2a_{110}xy + a_{020}y^2 + 2a_{011}yz + a_{002}z^2 + 2a_{101}xz\\ &\quad +2a_{100}x + 2a_{010}y + 2a_{001}z + a_{000} = 0,\end{aligned}$$

or

$$\boldsymbol{P}^T \mathcal{Q} \boldsymbol{P} = 0, \quad \text{where} \quad \mathcal{Q} = \begin{pmatrix} a_{200} & a_{110} & a_{101} & a_{100}\\ a_{110} & a_{020} & a_{011} & a_{010}\\ a_{101} & a_{011} & a_{002} & a_{001}\\ a_{100} & a_{010} & a_{001} & a_{000}\end{pmatrix}.$$

(Note that this is slightly different from the parameterization of quadric surfaces introduced in chapter 2. The parameterization used here facilitates the use of a constraint on the Frobenius form of $\mathcal{A}$.)
Suppose we observe n points with homogeneous coordinate vectors $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. To fit a quadric surface to these points, we minimize $\sum_{i=1}(\boldsymbol{P}_i^T \mathcal{Q} \boldsymbol{P}_i)^2$ with respect to the 10 coefficients defining the symmetric matrix $\mathcal{Q}$, under the constraint that $\mathcal{Q}$ has unit Frobenius norm.
This is equivalent to minimizing $|\mathcal{A}\boldsymbol{q}|^2$ under the constraint $|\boldsymbol{q}|^2 = 1$, where

$$\mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2y_1z_1 & z_1^2 & 2x_1z_1 & 2x_1 & 2y_1 & 2z_1 & 1\\ x_2^2 & 2x_2y_2 & y_2^2 & 2y_2z_2 & z_2^2 & 2x_2z_2 & 2x_2 & 2y_2 & 2z_2 & 1\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\ldots & \\ x_n^2 & 2x_ny_n & y_n^2 & 2y_nz_n & z_n^2 & 2x_nz_n & 2x_n & 2y_n & 2z_n & 1\end{pmatrix}$$

and $\boldsymbol{q} \stackrel{\text{def}}{=} (a_{200}, a_{110}, a_{020}, a_{011}, a_{002}, a_{101}, a_{100}, a_{010}, a_{001}, a_{000})^T$.
This is a homogeneous linear least-squares problem that can be solved using the eigenvalue/eigenvector methods described in chapter 3.

21.10. Show that a surface triangle maps onto a patch with hyperbolic edges in α, β space.

Solution Consider a triangle edge with extremities Q_1 and Q_2. Any point along this edge can be written as $Q = (1-t)Q_1 + tQ_2$. If the corresponding spin coordinates are α and β, we have

$$\beta = \overrightarrow{PQ}\cdot\boldsymbol{n} = [(1-t)\overrightarrow{PQ_1} + t\overrightarrow{PQ_2}]\cdot\boldsymbol{n} = (1-t)\beta_1 + t\beta_2,$$

where β_1 and β_2 are the β spin coordinates of Q_1 and Q_2.
Now we have

$$\alpha = |\overrightarrow{PQ}\times\boldsymbol{n}| = |[(1-t)\overrightarrow{PQ_1} + t\overrightarrow{PQ_2}]\times\boldsymbol{n}| = |(1-t)\boldsymbol{a}_1 + t\boldsymbol{a}_2|,$$

where $\boldsymbol{a}_1 = \overrightarrow{PQ_1} \times \boldsymbol{n}$ and $\boldsymbol{a}_2 = \overrightarrow{PQ_2} \times \boldsymbol{n}$. Computing t as a function of β and substituting in this expression for α yields

$$\alpha = |\frac{\beta_2 - \beta}{\beta_2 - \beta_1}\boldsymbol{a}_1 + \frac{\beta - \beta_1}{\beta_2 - \beta_1}\boldsymbol{a}_2|.$$

Squaring this equation and clearing the denominators yields

$$(\beta_2 - \beta_1)^2\alpha^2 - |\boldsymbol{a}_1 - \boldsymbol{a}_2|^2\beta^2 + \lambda\beta + \mu = 0,$$

where λ and μ are constants. This is indeed the equation of a hyperbola.

Programming Assignments

21.11. Implement molecule-based smoothing and the computation of principal directions and curvatures.

21.12. Implement the region-growing approach to plane segmentation described in this chapter.

21.13. Implement an algorithm for computing the lines of curvature of a surface from its range image. Hint: Use a curve-growing algorithm analogous to the region-growing algorithm for plane segmentation.

21.14. Implement the Besl–McKay ICP registration algorithm.

21.15. Marching squares in the plane: Develop and implement an algorithm for finding the zero set of a planar density function. Hint: Work out the possible ways a curve may intersect the edges of a pixel, and use linear interpolation along these edges to identify the zero set.

21.16. Implement the registration part of the Faugeras–Hebert algorithm.

CHAPTER 22

Finding Templates using Classifiers

PROBLEMS

22.1. Assume that we are dealing with measurements $\boldsymbol{x}$ in some feature space S. There is an open set D where any element is classified as class one, and any element in the interior of $S-D$ is classified as class two.

(a) Show that

$$\begin{aligned} R(s) &= Pr\left\{1 \rightarrow 2|\text{using } s\right\} L(1 \rightarrow 2) + Pr\left\{2 \rightarrow 1|\text{using } s\right\} L(2 \rightarrow 1) \\ &= \int_D p(2|\boldsymbol{x}) d\boldsymbol{x} L(1 \rightarrow 2) + \int_{S-D} p(1|\boldsymbol{x}) d\boldsymbol{x} L(2 \rightarrow 1). \end{aligned}$$

(b) Why are we ignoring the boundary of D (which is the same as the boundary of $S-D$) in computing the total risk?

Solution

(a) Straightforward.

(b) Boundary has measure zero.

22.2. In Section 22.2, we said that if each class-conditional density had the same covariance, the classifier of Algorithm 22.2 boiled down to comparing two expressions that are linear in $\boldsymbol{x}$.

(a) Show that this is true.

(b) Show that if there are only two classes, we need only test the sign of a linear expression in $\boldsymbol{x}$.

22.3. In Section 22.3.1, we set up a feature u, where the value of u on the ith data point is given by $u_i = \boldsymbol{v} \cdot (\boldsymbol{x}_i - \boldsymbol{\mu})$. Show that u has zero mean.

Solution The mean of $\boldsymbol{u}_i$ is the mean of $\boldsymbol{v} \cdot (\boldsymbol{x}_i - \mu)$ which is the mean of $\boldsymbol{v} \cdot \boldsymbol{x}_i$ minus $\boldsymbol{v} \cdot \mu$; but the mean of $\boldsymbol{v} \cdot \boldsymbol{x}_i$ is $\boldsymbol{v}$ dotted with the mean of $\boldsymbol{x}_i$, which mean is μ.

22.4. In Section 22.3.1, we set up a series of features u, where the value of u on the ith data point is given by $u_i = \boldsymbol{v} \cdot (\boldsymbol{x}_i - \boldsymbol{\mu})$. We then said that the $\boldsymbol{v}$ would be eigenvectors of Σ, the covariance matrix of the data items. Show that the different features are independent using the fact that the eigenvectors of a symmetric matrix are orthogonal.

22.5. In Section 22.2.1, we said that the ROC was invariant to choice of prior. Prove this.

Programming Assignments

22.6. Build a program that marks likely skin pixels on an image; you should compare at least two different kinds of classifier for this purpose. It is worth doing this carefully because many people have found skin filters useful.

22.7. Build one of the many face finders described in the text.

CHAPTER 23

Recognition by Relations Between Templates

CHAPTER 24

Geometric Templates from Spatial Relations

PROBLEMS

24.1. Defining a Brooks transform: Consider a 2D shape whose boundary is the curve Γ defined by $\boldsymbol{x} : I \rightarrow \mathbb{R}^2$ and parameterized by arc length. The line segment joining any two points $\boldsymbol{x}_1 \stackrel{\text{def}}{=} \boldsymbol{x}(s_1)$ and $\boldsymbol{x}_2 \stackrel{\text{def}}{=} \boldsymbol{x}(s_2)$ on Γ defines a cross-section of the shape, with length $l(s_1, s_2) = |\boldsymbol{x}_1 - \boldsymbol{x}_2|$. We can thus reduce the problem of studying the set of cross-sections of the shape to the study of the topography of the surface S associated with the height function $h : I^2 \rightarrow \mathbb{R}^+$ defined by $h(s_1, s_2) = \frac{1}{2} l(s_1, s_2)^2$. In this context, the ribbon associated with Γ can be defined (Ponce *et al.*, 1999) as the set of cross-sections whose end-points correspond to valleys of S, (i.e., according to Haralick, 1983 or Haralick, Watson, and Laffey, 1983, the set of pairs (s_1, s_2) where the gradient ∇h of h is an eigenvector of the Hessian $\mathcal{H}$, and where the eigenvalue associated with the other eigenvector of the Hessian is positive).
Let $\boldsymbol{u}$ denote the unit vector such that $\boldsymbol{x}_1 - \boldsymbol{x}_2 = l\boldsymbol{u}$. We denote by $\boldsymbol{t}_i$ the unit tangent in $\boldsymbol{x}_i$ ($i = 1, 2$), and by θ_i and κ_i, respectively, the angle between the vectors $\boldsymbol{u}$ and $\boldsymbol{t}_i$, and the curvature in $\boldsymbol{x}_i$. Show that the ribbon associated with Γ is the set of cross-sections of this shape whose endpoints satisfy

$$(\cos^2\theta_1 - \cos^2\theta_2)\cos(\theta_1 - \theta_2) + l\cos\theta_1\cos\theta_2(\kappa_1\sin\theta_1 + \kappa_2\sin\theta_2) = 0.$$

Solution If $\boldsymbol{v}$ is a unit vector such that $\boldsymbol{u}$ and $\boldsymbol{v}$ form a right-handed orthonornal coordinate system for the plane, we can write $\boldsymbol{t}_i = \cos\theta_i\boldsymbol{u} + \sin\theta_i\boldsymbol{v}$ and $\boldsymbol{n}_i = -\sin\theta_i\boldsymbol{u} + \cos\theta_i\boldsymbol{v}$ for $i = 1, 2$. The gradient of h is

$$\nabla h = \begin{pmatrix} \boldsymbol{t}_1 \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) \\ -\boldsymbol{t}_2 \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) \end{pmatrix} = l\begin{pmatrix} \cos\theta_1 \\ -\cos\theta_2 \end{pmatrix},$$

and its Hessian is

$$\begin{aligned} \mathcal{H} &= \begin{pmatrix} 1 + \kappa_1\boldsymbol{n}_1 \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) & -\boldsymbol{t}_1 \cdot \boldsymbol{t}_2 \\ -\boldsymbol{t}_1 \cdot \boldsymbol{t}_2 & 1 - \kappa_2\boldsymbol{n}_2 \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) \end{pmatrix} \\ &= \begin{pmatrix} 1 - l\kappa_1\sin\theta_1 & -\cos(\theta_1 - \theta_2) \\ -\cos(\theta_1 - \theta_2) & 1 + l\kappa_2\sin\theta_2 \end{pmatrix}. \end{aligned}$$

We now write that the gradient is an eigenvector of the Hessian, or if "$\times$" denotes the operator associating with two vectors in $\mathbb{R}^2$ the determinant of their coordinates,

$$\begin{aligned} \boldsymbol{0} &= (\mathcal{H}\nabla h) \times \nabla h \\ &= l^2\begin{pmatrix} (1 - l\kappa_1\sin\theta_1)\cos\theta_1 + \cos\theta_2\cos(\theta_1 - \theta_2) \\ -\cos\theta_1\cos(\theta_1 - \theta_2) - (1 + l\kappa_2\sin\theta_2)\cos\theta_2 \end{pmatrix} \times \begin{pmatrix} \cos\theta_1 \\ -\cos\theta_2 \end{pmatrix}, \end{aligned}$$

which simplifies immediately into

$$(\cos^2\theta_1 - \cos^2\theta_2)\cos(\theta_1 - \theta_2) + l\cos\theta_1\cos\theta_2(\kappa_1\sin\theta_1 + \kappa_2\sin\theta_2) = 0.$$

24.2. Generalized cylinders: The definition of a valley given in the previous exercise is valid for height surfaces defined over n-dimensional domains and valleys form curves in any dimension. Briefly explain how to extend the definition of ribbons given in that exercise to a new definition for generalized cylinders. Are difficulties not encountered in the two-dimensional case to be expected?

Solution Following the ideas from the previous exercise, it is possible to define the generalized cylinder associated with a volume V by the valleys of a height function defined over a three-dimensional domain: for example, we can pick some parameterization Π of the three-dimensional set of all planes by three parameters (s_1, s_2, s_3), and define $h(s_1, s_2, s_3)$ as the area of the region where V and the plane $\Pi(s_1, s_2, s_3)$ intersect. The valleys (and ridges) of this height function are characterized as before by $(\mathcal{H}\nabla h) \times \nabla h = \mathbf{0}$, where "$\times$" denotes this time the operator associating with two vectors their cross product. They form a one-dimensional set of cross-sections of V that can be taken as the generalized cylinder description of this volume.
There are some difficulties with this definition that are not encountered in the two-dimensional case: In particular, there is no natural parameterization of the cross-sections of a volume by the points on its boundary, and the valleys found using a plane parameterization depend on the choice of this parameterization. Moreover, the cross-section of a volume by a plane may consist of several connected components. See Ponce *et al.* (1999) for a discussion.

24.3. Skewed symmetries: A skewed symmetry is a Brooks ribbon with a straight axis and generators at a fixed angle θ from the axis. Skewed symmetries play an important role in line-drawing analysis because it can be shown that a bilaterally symmetric planar figure projects onto a skewed symmetry under orthographic projection (Kanade, 1981). Show that two contour points P_1 and P_2 forming a skewed symmetry verify the equation

$$\frac{\kappa_2}{\kappa_1} = -\left[\frac{\sin\alpha_2}{\sin\alpha_1}\right]^3,$$

where κ_i denotes the curvature of the skewed symmetry's boundary in P_i $(i = 1, 2)$, and α_i denotes the angle between the line joining the two points and the normal to this boundary (Ponce, 1990).
Hint: Construct a parametric representation of the skewed symmetry.

Solution Given two unit vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ separated by an angle of θ, we parameterize a skewed symmetry by

$$\begin{cases} \boldsymbol{x}_1 = s\boldsymbol{v} - r(s)\boldsymbol{u}, \\ \boldsymbol{x}_2 = s\boldsymbol{v} + r(s)\boldsymbol{u}, \end{cases}$$

where $\boldsymbol{u}$ is the generator direction, $\boldsymbol{v}$ is the skew axis direction, and $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote the two endpoints of the ribbon generators. Differentiating $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with respect to s yields

$$\begin{cases} \boldsymbol{x}_1' = \boldsymbol{v} - r'\boldsymbol{u}, \\ \boldsymbol{x}_2' = \boldsymbol{v} + r'\boldsymbol{u}, \end{cases}$$

and

$$\begin{cases} \boldsymbol{x}_1'' = -r''\boldsymbol{u}, \\ \boldsymbol{x}_2'' = r''\boldsymbol{u}. \end{cases}$$

Let us define α_i as the (unsigned) angle between the normal in $\boldsymbol{x}_i$ $(i = 1, 2)$ and the line joining the two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. We have

$$\begin{cases} \sin\alpha_1 = \dfrac{1}{|\boldsymbol{x}_1'|}|\boldsymbol{u} \times \boldsymbol{x}_1'| = \dfrac{\sin\theta}{\sqrt{1 - 2r'\cos\theta + r'^2}}, \\ \sin\alpha_2 = \dfrac{1}{|\boldsymbol{x}_2'|}|\boldsymbol{u} \times \boldsymbol{x}_2'| = \dfrac{\sin\theta}{\sqrt{1 + 2r'\cos\theta + r'^2}}, \end{cases}$$

where "$\times$" denotes the operator associating with two vectors in $\mathbb{R}^2$ the determinant of their coordinates.
Now remember from Ex. 19.4 that the curvature of a parametric curve is $\kappa = |\boldsymbol{x}' \times \boldsymbol{x}''|/|\boldsymbol{x}'|^3$. Using the convention that the curvature is positive when the ribbon boundary is convex, we obtain

$$\begin{cases} \kappa_1 = \dfrac{-\,r''\sin\theta}{(1 - 2r'\cos\theta + r'^2)^{3/2}}, \\ \kappa_2 = \dfrac{r''\sin\theta}{(1 + 2r'\cos\theta + r'^2)^{3/2}}, \end{cases}$$

and the result follows immediately.

Programming Assignments

24.4. Write an erosion-based skeletonization program. The program should iteratively process a binary image until it does not change anymore. Each iteration is divided into eight steps. In the first one, pixels from the input image whose 3×3 neighborhood matches the left pattern below (where "*" means that the corresponding pixel value does not matter) are assigned a value of zero in an auxiliary image; all other pixels in that picture are assigned their original value from the input image.

0	0	0
*	1	*
1	1	1

0	0	*
0	1	1
*	1	1

The auxiliary picture is then copied into the input image, and the process is repeated with the right pattern. The remaining steps of each iteration are similar and use the six patterns obtained by consecutive 90-degree rotations of the original ones. The output of the program is the 4-connected skeleton of the original region (Serra, 1982).

24.5. Implement the FORMS approach to skeleton detection.

24.6. Implement the Brooks transform.

24.7. Write a program for finding skewed symmetries. You can implement either (a) a naive $O(n^2)$ algorithm comparing all pairs of contour points, or (b) the $O(kn)$ projection algorithm proposed by Nevatia and Binford (1977). The latter method can be summarized as follows: Discretize the possible orientations of local ribbon axes; for each of these k directions, project all contour points into buckets and verify the local skewed symmetry condition for points within the same bucket only; finally, group the resulting ribbon pairs into ribbons.

CHAPTER 25

Application: Finding in Digital Libraries

CHAPTER 26

Application: Image-Based Rendering

PROBLEMS

26.1. Given $n+1$ points $P_0, \dots, P_n$, we recursively define the parametric curve $P_i^k(t)$ by $P_i^0(t) = P_i$ and

$$P_i^k(t) = (1-t)P_i^{k-1}(t) + tP_{i+1}^{k-1}(t) \quad \text{for} \quad k = 1, \dots, n \quad \text{and} \quad i = 0, \dots, n-k.$$

We show in this exercise that $P_0^n(t)$ is the Bézier curve of degree n associated with the $n+1$ points $P_0, \dots, P_n$. This construction of a Bézier curve is called the *de Casteljeau algorithm*.

(a) Show that Bernstein polynomials satisfy the recursion

$$b_i^{(n)}(t) = (1-t)b_i^{(n-1)}(t) + tb_{i-1}^{(n-1)}(t)$$

with $b_0^{(0)}(t) = 1$ and, by convention, $b_j^{(n)}(t) = 0$ when $j < 0$ or $j > n$.

(b) Use induction to show that

$$P_i^k(t) = \sum_{j=0}^{k} b_j^{(k)}(t) P_{i+j} \quad \text{for} \quad k = 0, \dots, n \quad \text{and} \quad i = 0, \dots, n-k.$$

Solution Let us recall that the Bernstein polynomials of degree n are defined by $b_i^{(n)}(t) \stackrel{\text{def}}{=} \binom{n}{i} t^i (1-t)^{n-i}$ $(i = 0, \dots, n)$.

(a) Writing

$$\begin{aligned}(1-t)b_i^{(n-1)}(t) + tb_{i-1}^{(n-1)}(t) &= \tbinom{n-1}{i} t^i (1-t)^{n-i} + \tbinom{n-1}{i-1} t^i (1-t)^{n-i} \\ &= \frac{(n-1)!}{(n-i)!i!}[(n-i)+i] t^i (1-t)^{n-i} = b_i^{(n)}(t).\end{aligned}$$

shows that the recursion is satisfied when $i > 0$ and $i < n$. It also holds when $i = 0$ since, by definition, $b_0^{(n)}(t) = (1-t)^n = (1-t)b_0^{(n-1)}(t)$ and, by convention, $b_{-1}^{(n-1)}(t) = 0$. Likewise, the recursion is satisfied when $i = n$ since, by definition, $b_n^{(n)}(t) = t^n = tb_{n-1}^{(n-1)}(t)$ and, by convention, $b_n^{(n-1)}(t) = 0$.

(b) The induction hypothesis is obviously true for $k = 0$ since, by definition, $P_i^0(t) = P_i$ for $i = 0, \dots, n$. Suppose it is true for $k = l-1$. We have, by definition,

$$P_i^l(t) = (1-t)P_i^{l-1}(t) + tP_{i+1}^{l-1}(t).$$

Thus, according to the induction hypothesis,

$$\begin{aligned}P_i^l(t) &= (1-t)\textstyle\sum_{j=0}^{l-1} b_j^{(l-1)}(t)P_{i+j} + t\sum_{j=0}^{l-1} b_j^{(l-1)}(t)P_{i+1+j} \\ &= (1-t)\textstyle\sum_{j=0}^{l-1} b_j^{(l-1)}(t)P_{i+j} + t\sum_{m=1}^{l} b_{m-1}^{(l-1)}(t)P_{i+m},\end{aligned}$$

where we have made the change of variables $m = j+1$ in the second summation. Using (a) and the convention that $b_j^{(n)}(t) = 0$ when $j < 0$ or $j > n$, we obtain

$$P_i^l(t) = \sum_{j=0}^{l} [(1-t)b_j^{(l-1)}(t) + tb_{j-1}^{(l-1)}(t)]P_{i+j} = \sum_{j=0}^{l} b_j^{(l)}(t)P_{i+j},$$

which concludes the inductive proof. In particular, picking $k = n$ and $i = 0$ shows that $P_0^n(t)$ is the the Bézier curve of degree n associated with the $n+1$ points $P_0, \ldots, P_n$.

26.2. Consider a Bézier curve of degree n defined by $n+1$ control points $P_0, \ldots, P_n$. We address here the problem of constructing the $n+2$ control points $Q_0, \ldots, Q_{n+1}$ of a Bézier curve of degree $n+1$ with the same shape. This process is called *degree elevation*. Show that $Q_0 = P_0$ and

$$Q_j = \frac{j}{n+1}P_{j-1} + (1 - \frac{j}{n+1})P_j \quad \text{for} \quad j = 1, \ldots, n+1.$$

Hint: Write that the same point is defined by the barycentric combinations associated with the two curves, and equate the polynomial coefficients on both sides of the equation.

Solution We write

$$P(t) = \sum_{j=0}^{n} \binom{n}{j} t^j (1-t)^{n-j} P_j = \sum_{j=0}^{n+1} \binom{n+1}{j} t^j (1-t)^{n+1-j} Q_j.$$

To equate the polynomial coefficients of both expressions for $P(t)$, we multiply each Bernstein polynomial in the first expression by $t+1-t=1$. With the usual change of variables in the second line below, this yields

$$\begin{aligned} P(t) &= \textstyle\sum_{j=0}^{n} \binom{n}{j} t^{j+1}(1-t)^{n-j}P_j + \sum_{j=0}^{n} \binom{n}{j} t^j(1-t)^{n+1-j}P_j \\ &= \textstyle\sum_{k=1}^{n+1} \binom{n}{k-1} t^k(1-t)^{n+1-k}P_{k-1} + \sum_{j=0}^{n} \binom{n}{j} t^j(1-t)^{n+1-j}P_j \\ &= (1-t)^{n+1}P_0 + t^{n+1}P_n + \textstyle\sum_{j=1}^{n} [t^j(1-t)^{n+1-j}][\binom{n}{j-1}P_{j-1} + \binom{n}{j}P_j] \end{aligned}$$

Note that $P(0)$ is the first control point of both arcs, so $P_0 = Q_0$. Likewise, $P(1)$ is the last control point, so $P_n = Q_{n+1}$. This is confirmed by examining the polynomial coefficients corresponding to $j = 0$ and $j = n+1$ in the two expressions of $P(t)$. Equating the remaining coefficients yields

$$\binom{n+1}{j} Q_j = \binom{n}{j-1} P_{j-1} + \binom{n}{j} P_j \quad \text{for} \quad j = 1, \ldots, n,$$

or

$$\frac{(n+1)!}{(n+1-j)!j!} Q_j = \frac{n!}{(n+1-j)!(j-1)!} P_{j-1} + \frac{n!}{(n-j)!j!} P_j.$$

This can finally be rewritten as

$$Q_j = \frac{j}{n+1}P_{j-1} + \frac{n+1-j}{n+1}P_j = \frac{j}{n+1}P_{j-1} + (1 - \frac{j}{n+1})P_j.$$

Note that this is indeed a barycentric combination, which justifies our calculations, and that this expression is valid for $j > 0$ and $j < n+1$. It is in fact also valid for $j = n+1$ since, as noted before, we have $P_n = Q_{n+1}$.

26.3. Show that the tangent to the Bézier curve $P(t)$ defined by the $n+1$ control points $P_0, \dots, P_n$ is

$$P'(t) = n \sum_{j=0}^{n-1} b_j^{(n-1)}(t)(P_{j+1} - P_j).$$

Conclude that the tangents at the endpoints of a Bézier arc are along the first and last line segments of its control polygon.

Solution Writing the tangent as

$$\begin{aligned}
P'(t) &= \sum_{j=0}^{n} b_j^{(n)\prime}(t) P_j \\
&= \sum_{j=1}^{n} \binom{n}{j} j t^{j-1}(1-t)^{n-j} P_j + \sum_{j=0}^{n-1} \binom{n}{j}(n-j) t^j (1-t)^{n-j-1} P_j \\
&= n \sum_{j=1}^{n} \binom{n-1}{j-1} t^{j-1}(1-t)^{n-j} P_j + n \sum_{j=0}^{n-1} \binom{n-1}{j} t^j (1-t)^{n-j-1} P_j \\
&= n \sum_{k=0}^{n-1} \binom{n-1}{k} t^k (1-t)^{n-k-1} P_{k+1} + n \sum_{j=0}^{n-1} \binom{n-1}{j} t^j (1-t)^{n-j-1} P_j \\
&= n \sum_{j=0}^{n-1} \binom{n-1}{j} t^j (1-t)^{n-1-1}(P_{j+1} - P_j) \\
&= n \sum_{j=0}^{n-1} b_j^{(n-1)}(t)(P_{j+1} - P_j)
\end{aligned}$$

proves the result (note the change of variable $k = j-1$ in the fourth line of the equation).
The tangents at the endpoints of the arc correspond to $t = 0$ and $t = 1$. Since $b_j^{(n-1)}(0) = 0$ for $j > 0$, $b_j^{(n-1)}(1) = 0$ for $j < n-1$, $b_0^{(n-1)}(0) = 1$, and $b_{n-1}^{(n-1)}(1) = 1$, we conclude that

$$P'(0) = n(P_1 - P_0) \quad \text{and} \quad P'(1) = n(P_n - P_{n-1}),$$

which shows that the tangents at the endpoints of the Bézier arc along the first and last line segments of its control polygon.

26.4. Show that the construction of the points Q_i in Section 26.1.1 places these points in a plane that passes through the centroid O of the points C_i

Solution First it is easy to show that the points Q_i are indeed barycentric combinations of the points C_j: This follows immediately from the fact that, due to the regular and symmetric sampling of angles in the linear combination defining Q_i, the sum of the cosine terms is zero. Now let us shows that the points Q_i are coplanar, and more precisely, that any of these points can be written as a barycentric combination of the points

$$\left\{ \begin{aligned}
Q_1 &= \sum_{j=1}^{p} \tfrac{1}{p} \left\{ 1 + \cos\frac{\pi}{p} \cos\left([2(j-1)-1]\frac{\pi}{p} \right) \right\} C_j, \\
Q_{p-1} &= \sum_{j=1}^{p} \tfrac{1}{p} \left\{ 1 + \cos\frac{\pi}{p} \cos\left([2(j+1)-1]\frac{\pi}{p} \right) \right\} C_j, \\
Q_p &= \sum_{j=1}^{p} \tfrac{1}{p} \left\{ 1 + \cos\frac{\pi}{p} \cos\left([2j-1]\frac{\pi}{p} \right) \right\} C_j.
\end{aligned} \right.$$

We write

$$Q_i = \sum_{j=1}^{p} \frac{1}{p} \left\{ 1 + \cos\frac{\pi}{p} \cos\left([2(j-i)-1]\frac{\pi}{p} \right) \right\} C_j = aQ_1 + bQ_2 + cQ_3,$$

with $a+b+c=1$. If $\theta_{ij}=[2(j-i)-1]\pi/p$, this equation can be rewritten as

$$\begin{aligned}
\cos\theta_{ij} &= a\cos[2(j-1)-1]\frac{\pi}{p}+b\cos[2(j+1)-1]\frac{\pi}{p}+c\cos[2j-1]\frac{\pi}{p}\\
&= a\cos[\theta_{ij}+2(i-1)]\frac{\pi}{p}+b\cos[\theta_{ij}+2(i+1)]\frac{\pi}{p}+c\cos[\theta_{ij}+2i]\frac{\pi}{p}\\
&= \cos\theta_{ij}\left\{a\cos[2(i-1)\frac{\pi}{p}]+b\cos[2(i+1)\frac{\pi}{p}]+c\cos[2i\frac{\pi}{p}]\right\}\\
&- \sin\theta_{ij}\left\{a\sin[2(i-1)\frac{\pi}{p}]+b\sin[2(i+1)\frac{\pi}{p}]+c\sin[2i\frac{\pi}{p}]\right\},
\end{aligned}$$

or equivalently, adding the constraint that a, b, and c must add to 1:

$$\begin{pmatrix}\cos[2(i-1)\frac{\pi}{p}] & \cos[2(i+1)\frac{\pi}{p}] & +\cos[2i\frac{\pi}{p}]\\ \sin[2(i-1)\frac{\pi}{p}] & \sin[2(i+1)\frac{\pi}{p}] & +\sin[2i\frac{\pi}{p}]\\ 1 & 1 & 1\end{pmatrix}\begin{pmatrix}a\\b\\c\end{pmatrix}=\begin{pmatrix}1\\0\\1\end{pmatrix}.$$

This system of three equations in three unknowns admits (in general) a unique solution, which shows that any Q_i can be written as a barycentric combination of the points Q_1, Q_{p-1}, and Q_p, and that the points Q_1, Q_2, ..., Q_p are indeed coplanar.
Now, it is easy to see that the points Q_i can be written as

$$\begin{cases}Q_1=\lambda_1C_1+\lambda_2C_2+\ldots+\lambda_pC_p\\ Q_2=\lambda_pC_1+\lambda_1C_2+\ldots+\lambda_{p-1}C_p\\ \ldots\\ Q_p=\lambda_2C_1+\lambda_3C_2+\ldots+\lambda_1C_p\end{cases}$$

with $\lambda_1+\ldots+\lambda_p=1$, and it follows immediately that the centroid of the points Q_i (which obviously belongs to the plane spanned by these points) is also the centroid of the points C_i.

26.5. Façade's photogrammetric module. We saw in the exercises of chapter 3 that the mapping between a line δ with Plücker coordinate vector $\mathbf{\Delta}$ and its image δ with homogeneous coordinates $\boldsymbol{\delta}$ can be represented by $\rho\boldsymbol{\delta}=\tilde{\mathcal{M}}\mathbf{\Delta}$. Here, $\mathbf{\Delta}$ is a function of the model parameters, and $\tilde{\mathcal{M}}$ depends on the corresponding camera position and orientation.

(a) Assuming that the line δ has been matched with an image edge e of length l, a convenient measure of the discrepancy between predicted and observed data is obtained by multiplying by l the mean squared distance separating the points of e from δ. Defining $d(t)$ as the signed distance between the edge point $p=(1-t)p_0+tp_1$ and the line δ, show that

$$E=\int_0^1 d^2(t)dt=\frac{1}{3}(d(0)^2+d(0)d(1)+d(1)^2).$$

where d_0 and d_1 denote the (signed) distances between the endpoints of e and δ.

(b) If $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ denote the homogeneous coordinate vectors of these points, show that

$$d_0=\frac{1}{|[\tilde{\mathcal{M}}\mathbf{\Delta}]_2|}\boldsymbol{p}_0^T\tilde{\mathcal{M}}\mathbf{\Delta}\quad\text{and}\quad d_1=\frac{1}{|[\tilde{\mathcal{M}}\mathbf{\Delta}]_2|}\boldsymbol{p}_1^T\tilde{\mathcal{M}}\mathbf{\Delta},$$

where $[\boldsymbol{a}]_2$ denotes the vector formed by the first two coordinates of the vector $\boldsymbol{a}$ in $\mathbb{R}^3$

(c) Formulate the recovery of the camera and model parameters as a non-linear least-squares problem.

Solution

(a) Let us write the equation of δ as $\boldsymbol{n} \cdot \boldsymbol{p} = D$, where $\boldsymbol{n}$ is a unit vector and D is the distance between the origin and δ. The (signed) distance between the point $p = (1-t)p_0 + tp_1$ and δ is

$$d(t) = \boldsymbol{n} \cdot \boldsymbol{p} - D = \boldsymbol{n} \cdot [(1-t)\boldsymbol{p}_0 + t\boldsymbol{p}_1] - [(1-t)+t]D = (1-t)d(0) + td(1).$$

We have therefore

$$\begin{aligned} E &= \int_0^1 d^2(t)dt = \int_0^1 [(1-t)^2 d(0)^2 + 2(1-t)td(0)d(1) + t^2 d(1)^2]dt \\ &= \left[-\tfrac{1}{3}(1-t)^3\right]_0^1 d(0)^2 + \left[t^2 - \tfrac{2}{3}t^3\right]_0^1 d_0 d_1 + \left[\tfrac{1}{3}t^3\right]_0^1 d(1)^2 \\ &= \tfrac{1}{3}(d(0)^2 + d(0)d(1) + d(1)^2). \end{aligned}$$

(b) With the same notation as before, we can write $\boldsymbol{\delta}^T = (\boldsymbol{n}^T, D)^T$. Since $\rho\boldsymbol{\delta} = \tilde{\mathcal{M}}\boldsymbol{\Delta}$ and $\boldsymbol{n}$ is a unit vector, it follows immediately that

$$d(0) = \boldsymbol{n}\cdot\boldsymbol{p}_0 - D = \frac{1}{|[\tilde{\mathcal{M}}\boldsymbol{\Delta}]_2|}\boldsymbol{p}_0^T\tilde{\mathcal{M}}\boldsymbol{\Delta} \text{ and } d(1) = \boldsymbol{n}\cdot\boldsymbol{p}_1 - D = \frac{1}{|[\tilde{\mathcal{M}}\boldsymbol{\Delta}]_2|}\boldsymbol{p}_1^T\tilde{\mathcal{M}}\boldsymbol{\Delta}.$$

(c) Given n matches established across m images between the lines Δ_j ($j = 1, \ldots, n$) and the corresponding image edges e_j with endpoints p_{j0} and p_{j1} and lengths l_j, we can formulate the recovery of the camera parameters $\mathcal{M}_i$ and the model parameters associated with the line coordinate vectors $\boldsymbol{\Delta}_j$ as the mimimization of the mean-squared error $\frac{1}{mn}\sum_{i=1}^m \sum_j^n \frac{l_j}{3} f_{ij}^2$, where

$$f_{ij} \stackrel{\text{def}}{=} \frac{1}{|[\tilde{\mathcal{M}}_i\boldsymbol{\Delta}_j]_2|}\sqrt{(\boldsymbol{p}_{j0}^T\tilde{\mathcal{M}}_i\boldsymbol{\Delta}_j)^2 + (\boldsymbol{p}_{j0}^T\tilde{\mathcal{M}}_i\boldsymbol{\Delta}_j)(\boldsymbol{p}_{j1}^T\tilde{\mathcal{M}}_i\boldsymbol{\Delta}_j) + (\boldsymbol{p}_{j1}^T\tilde{\mathcal{M}}_i\boldsymbol{\Delta}_j)^2},$$

with respect to the unknown parameters (note that the term under the radical is positive since it is equal—up to a positive constant—to the integral of d^2). It follows that the recovery of these parameters can be expressed as a (non-linear) least-squares problem.

26.6. Show that a basis for the eight-dimensional vector space V formed by all affine images of a fixed set of points $P_0, \ldots, P_{n-1}$ can be constructed from at least two images of these points when $n \geq 4$.
Hint: Use the matrix

$$\begin{pmatrix} u_0^{(1)} & v_0^{(1)} & \ldots & u_0^{(m)} & v_0^{(m)} \\ \ldots & \ldots & \ldots & & \\ u_{n-1}^{(1)} & v_{n-1}^{(1)} & \ldots & u_{n-1}^{(m)} & v_{n-1}^{(m)} \end{pmatrix},$$

where $(u_i^{(j)}, v_i^{(j)})$ are the coordinates of the projection of the point P_i into image number j.

Solution The matrix introduced here is simply the transpose of the data matrix used in the Tomasi-Kanade factorization approach of chapter 12. With at least two views of $n \geq 4$ points, the singular value decomposition of this matrix can be used to estimate the points $\boldsymbol{P}_i$ $(i = 0, \ldots, n-1)$ and construct the matrix

$$\begin{pmatrix} \boldsymbol{P}_0^T & \boldsymbol{0}^T & 1 & 0 \\ \boldsymbol{0}^T & \boldsymbol{P}_0^T & 0 & 1 \\ \ldots & \ldots & \ldots & \ldots \\ \boldsymbol{P}_{n-1}^T & \boldsymbol{0}^T & 1 & 0 \\ \boldsymbol{0}^T & \boldsymbol{P}_{n-1}^T & 0 & 1 \end{pmatrix}$$

whose columns span the eight-dimensional vector space V formed by all images of these n points.

26.7. Show that the set of all projective images of a fixed scene is an eleven-dimensional variety.

Solution Writing

$$\begin{pmatrix} \boldsymbol{p}_0 \\ \ldots \\ \boldsymbol{p}_{n-1} \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_0}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_0} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_0}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_0} \\ \ldots \\ \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_0}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_0} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_0}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_0} \end{pmatrix}$$

shows that the set of all images of a fixed scene forms a surface embedded in $\mathbb{R}^{2n}$ and defined by rational equations in the row vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ of the projection matrix. Rational parametric surfaces are varieties whose dimension is given by the number of independent parameters. Since projection matrices are only defined up to scale, the dimension of the variety formed by all projective images of a fixed scene is 11.

26.8. Show that the set of all perspective images of a fixed scene (for a camera with constant intrinsic parameters) is a six-dimensional variety.

Solution A perspective projection matrix can always be written as $\rho\mathcal{M} = \mathcal{K}\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\end{pmatrix}$, where $\mathcal{K}$ is the matrix of intrinsic parameters, $\mathcal{R}$ is a rotation matrix, and ρ is a scalar accounting for the fact that $\mathcal{M}$ is only defined up to a scale factor. The matrix $\mathcal{A}$ formed by the three leftmost columns of $\mathcal{M}$ must therefore satisfy the five polynomial constraints associated with the fact that the columns of the matrix $\mathcal{K}^{-1}\mathcal{A}$ are orthogonal to each other and have the same length. Since the set of all projective images of a fixed scene is a variety of dimension 11, it follows that the set of all perspective images is a sub-variety of dimension $11 - 5 = 6$.

26.9. In this exercise, we show that Eq. (26.7) only admits two solutions.

(a) Show that Eq. (26.6) can be rewritten as

$$\begin{cases} X^2 - Y^2 + e_1 - e_2 = 0, \\ 2XY + e = 0, \end{cases} \tag{26.1}$$

where

$$\begin{cases} X = u + \alpha u_1 + \beta u_2, \\ Y = v + \alpha v_1 + \beta v_2, \end{cases}$$

and e, e_1, and e_2 are coefficients depending on u_1, v_1, u_2, v_2 and the structure parameters.

(b) Show that the solutions of Eq. (26.8) are given by

$$\left\{\begin{array}{l} X' = \sqrt[4]{(e_1-e_2)^2+e^2}\cos(\frac{1}{2}\arctan(e,e_1-e_2)), \\ Y' = \sqrt[4]{(e_1-e_2)^2+e^2}\sin(\frac{1}{2}\arctan(e,e_1-e_2)), \end{array}\right.$$

and $(X'',Y'') = (-X',-Y')$. (Hint: Use a change of variables to rewrite Eq. (26.8) as a system of trigonometric equations.)

Solution Recall that Eq. (26.6) has the form

$$\left\{\begin{array}{l} \boldsymbol{u}^T\mathcal{R}\boldsymbol{u} - \boldsymbol{v}^T\mathcal{R}\boldsymbol{v} = 0, \\ \boldsymbol{u}^T\mathcal{R}\boldsymbol{v} = 0, \end{array}\right. \quad \text{where} \quad \mathcal{R} = \begin{pmatrix} (1+\lambda^2)z^2+\alpha^2 & \lambda\mu z^2+\alpha\beta & \alpha \\ \lambda\mu z^2+\alpha\beta & \mu^2z^2+\beta^2 & \beta \\ \alpha & \beta & 1 \end{pmatrix}.$$

(a) Let us define the vector $\boldsymbol{\alpha} = (\alpha,\beta,1)^T$ and note that $X = \boldsymbol{\alpha}\cdot\boldsymbol{u}$ and $Y = \boldsymbol{\alpha}\cdot\boldsymbol{v}$. This allows us to write

$$\mathcal{R} = \boldsymbol{\alpha}\boldsymbol{\alpha}^t + z^2\begin{pmatrix} \mathcal{L} & 0 \\ \boldsymbol{0}^T & 0 \end{pmatrix} \quad \text{where} \quad \mathcal{L} = \begin{pmatrix} 1+\lambda^2 & \lambda\mu \\ \lambda\mu & \mu^2 \end{pmatrix}.$$

If $\boldsymbol{u}_2 = (u_1,u_2)^T$ and $\boldsymbol{v}_2 = (v_1,v_2)^T$, it follows that we have indeed

$$\left\{\begin{array}{l} 0 = \boldsymbol{u}^T\mathcal{R}\boldsymbol{u} - \boldsymbol{v}^T\mathcal{R}\boldsymbol{v} = X^2 - Y^2 + e_1 - e_2 = 0, \\ 0 = \boldsymbol{u}^T\mathcal{R}\boldsymbol{v} = \dfrac{1}{2}[2XY+e] = 0, \end{array}\right.$$

where $e_1 = z^2\boldsymbol{u}_2^T\mathcal{L}\boldsymbol{u}_2$, $e_2 = z^2\boldsymbol{v}_2^T\mathcal{L}\boldsymbol{v}_2$, and $e = 2z^2\boldsymbol{u}_2^T\mathcal{L}\boldsymbol{v}_2$.

(b) We can always write $X = a\cos\theta$ and $Y = a\sin\theta$ for some $a > 0$ and $\theta \in [0,2\pi]$. This allows us to rewrite rewrite Eq. (26.8) as

$$\left\{\begin{array}{l} a^2\cos 2\theta = e_2 - e_1 \\ a^2\sin 2\theta = -e \end{array}\right.$$

If follows that $a^4 = (e_1-e_2)^2 + e^2$ and, if $\phi = \arctan(e, e_1-e_2)$, the two solutions for θ are $\theta' = \phi/2$ and $\theta'' = \theta' + \pi$. Thus the solutions of (26.8) are $(X',Y') = (a\cos\theta', a\sin\theta')$ and $(X'',Y'') = (a\cos\theta'', a\sin\theta'') = -(X',Y')$.