COMPSCI.773 S1 T
Vision Guided Control
Lecture Notes – 2005
Part 1: Basics of Mono and Stereo Vision

P. Delmas, G. Gimel'farb
CITR, Tamaki Campus
Department of Computer Science
University of Auckland

E-mail: {pdel016,ggim001}@cs.auckland.ac.nz

Preface

Computer or machine vision pursues the goal of describing and understanding natural 3D scenes using one or more 2D images. Vision guided control in industrial automation or robotics based on image acquisition and understanding but involves specific requirements such as (*i*) low cost, (*ii*) reliable operation, (*iii*) fundamental simplicity, (*iv*) real-time image analysis, and (*v*) easiness of scene illumination. These requirements are often diametrically opposed to many known computer vision results.

Table 1: Today's industrial and robotic vision systems.

Manufacturer	Country	Application areas
Adept Technology	USA	robot control, inspection
Cognex Corp.	USA	inspection, assembling
DS GmbH	Germany	industrial vision software
DVT	Canada	inspection, automation
Evolution Robotics	USA	mobile robotics
Imagis	Canada	security (face recognition)
ISRA Vision	USA	loading / unloading (2D/3D vision)
Keyence	USA	inspection
KLA - Tencor	USA	inspection
Machine Vision Products	USA	assembling
MobilEye	Israel	car cruise control
Newton Research Labs	USA	mobile robotics
Robotic Vision Systems	USA	inspection, assembling
Viisage	USA	security (face recognition)

From the very beginning of the research domain of computer vision and image analysis, the prospect of vision-guided control has been "just around the corner". After almost four decades of the world-wide efforts, that "corner" is still ahead and no really satisfactory general-purpose machine vision system has been developed. Nonetheless starting from the nineteen eighties, various commercial and experimental special-purpose machine vision systems have appeared, e.g. a "Videospray" system for paint spraying (Hayden Drysys International, UK), VS-100 vision system (Machine Intelligence Corp., USA) for industrial robots PUMA (Unimation Inc., USA), "Consight-I" structured-light industrial vision system (General Motors of Canada, Canada), and so on. But only a few of them have been justified by several years of operating experience in industrial environment. Some most known at present industrial firms producing machine vision systems are listed in Table 1 . For more information, see the Computer Vision industry web page

of Prof. David Lowe, University of Brithish Columbia, Canada:

http://www.cs.ubc.ca/spider/lowe/vision.html

Part 1 of these lecture notes covers in brief topics on mono and stereo image acquisition, 2D/3D vision geometry, camera calibration, colour detection and classification, and binary image vision (quantisation and segmentation). More details can be found in many available books and journal articles on computer and machine vision. These books and articles use different notation for the same quantities, so that it is little wonder that our notation below may differ from the one more familiar to you. Sometimes, when this creates no difficulties, the same character may denote different quantities, but in any case the notation involved is explicitly explained in each chapter.

Contents

Chapter 1

2D/3D Vision Geometry

1.1 Homogeneous coordinates

Not only a projective transformation of 3D points to a 2D plane, but also the simplest translations of 2D or 3D points are non-linear in terms of their Cartesian co-ordinates. To be able to represent basic 2D and 3D geometric transformations such as translations, rotations, scaling, and projections in mathematically convenient linear (i.e. matrix – vector) form, so-called **homogeneous coordinates** of 3D and 2D points are involved.

Below vectors and matrices are boldfaced and their transpositions are indicated by superscripts $^{\mathrm{T}}$ or $^{\mathsf{T}}$. We assume only column vectors but, for convenience sake, represent them sometimes as transposed row vectors, e.g. $(x, y)^{\mathsf{T}} \equiv \begin{pmatrix} x \\ y \end{pmatrix}$.

Each 2D point $\mathbf{p} = [x, y]^{\mathsf{T}}$ or 3D point $\mathbf{P} = [X, Y, Z]^{\mathsf{T}}$ can be represented in the homogeneous coordinates by the following 3- or 4-component vectors $(px, py, p)^{\mathsf{T}}$ and $(PX, PY, PZ, P)^{\mathsf{T}}$, respectively where t and T are arbitrary scalar factors. Each homogeneous vector is converted to the initial 2D or 3D Cartesian co-ordinate vector by dividing the first two or three components by the last one: $(h_1, h_2, h_3)^{\mathsf{T}} \rightarrow (x = h_1/h_3, y = h_2/h_3)$ or $(h_1, h_2, h_3, H_4)^{\mathsf{T}} \rightarrow (x = h_1/h_4, y = h_2/h_4, z = h_3/h_4)$.

Example: the 3D point $(5, 3, 2)^{\mathsf{T}}$ has the homogeneous representation $(5\tau, 3\tau, 2\tau, \tau)^{\mathsf{T}}$ with an arbitrary factor $\tau \neq 0$, e.g., $(5, 3, 2, 1)^{\mathsf{T}}$, or $(15, 9, 6, 3)^{\mathsf{T}}$, or $(-55, -33, -22, -11)^{\mathsf{T}}$ and so on. Conversely, the homogeneous vector $(30, 10, 15, 5)^{\mathsf{T}}$ represents the point $(6, 2, 3)$.

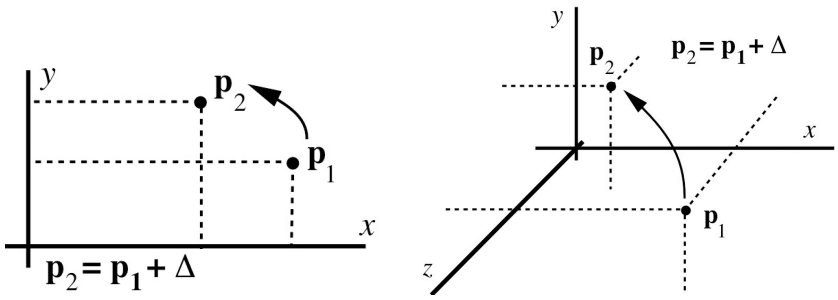
In homogeneous coordinates projective transformations as well as affine transformations (e.g. translations, rotations, scaling) are specified by linear equations.

Translation of a 2D point $\mathbf{p}_1 = (x_1, y_1)^{\mathsf{T}}$ to a new position $\mathbf{p}_2 = (x_2 + \delta_x, y_2 + \delta_y)^{\mathsf{T}} \equiv \mathbf{p}_1 + \boldsymbol{\Delta}$ where $\boldsymbol{\Delta} = (\delta_x, \delta_y)^{\mathsf{T}}$ is represented in homogeneous coordinates as

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

Translation of a 3D point $\mathbf{p}_1 = (x_1, y_1, z_1)^\mathsf{T}$ to position $\mathbf{p}_2 = (x_2 = x1 + \delta_x, y_2 + \delta_y, z_2 = z_1 + \delta_z)^\mathsf{T} \equiv \mathbf{p}_2 = \mathbf{p}_1 + \mathbf{\Delta}$ where $\mathbf{\Delta} = (\delta_x, \delta_y, \delta_z)^\mathsf{T}$ has a very similar representation in homogeneous coordinates :

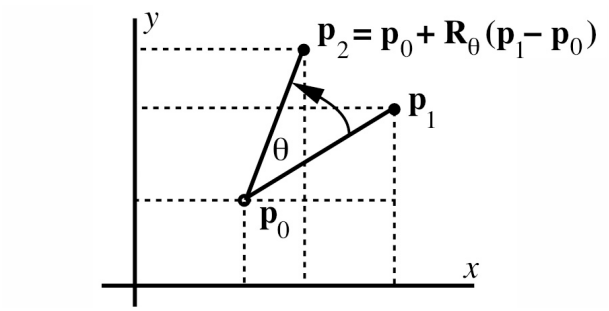
$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \delta_x \\ 0 & 1 & 0 & \delta_y \\ 0 & 0 & 1 & \delta_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$



Rotation of a 2D point $\mathbf{p}_1$ to a counter-clockwise (or left) angle θ around a given center $\mathbf{p}_0$ is represented in homogeneous coordinates as:

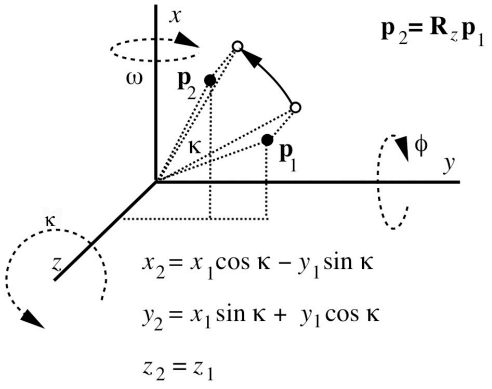
$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \left(\begin{array}{cc|c} & & \delta_x \\ \multicolumn{2}{c|}{\mathbf{R}_\theta} & \delta_y \\ \hline 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

where $\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ is the 2×2 rotation matrix and the offsets δ_x and δ_y are as follows in accord with the diagram below: $\delta_x = x_0(1 - \cos\theta) + y_0 \sin\theta$ and $\delta_y = -x_0 \sin\theta + y_0(1 - \cos\theta)$.



Rotation of a 3D point is specified with three angles, namely, swing (κ), pan (ϕ), and tilt (ω) angles of rotation around the z-, y-, and x-axis, respectively. In the general case, the 3D rotation around the co-ordinate origin of the left-hand co-ordinate frame is

decomposed to three successive rotations around the coordinate axes x, y, z, the rotation matrix being $\mathbf{R}_{\kappa,\phi,\omega} = \mathbf{R}_z(\kappa)\mathbf{R}_y(\phi)\mathbf{R}_x(\omega)$ where $\mathbf{R}_z(\kappa) = \begin{pmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}$ for the rotation from y- to x-axis, $\mathbf{R}_y(\phi) = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}$ for the rotation from z- to x-axis, and $\mathbf{R}_x(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{pmatrix}$ for the rotation from y- to z-axis.



The following resulting matrix $\mathbf{R}_{\kappa,\phi,\omega} =$

$$\begin{pmatrix} \cos\phi\cos\kappa & \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa & -\cos\omega\sin\phi\cos\kappa + \sin\omega\sin\kappa \\ -\cos\phi\sin\kappa & -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa & \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \\ \sin\phi & -\sin\omega\cos\phi & \cos\omega\cos\phi \end{pmatrix}$$

defines such a rotation in homogeneous coordinates:

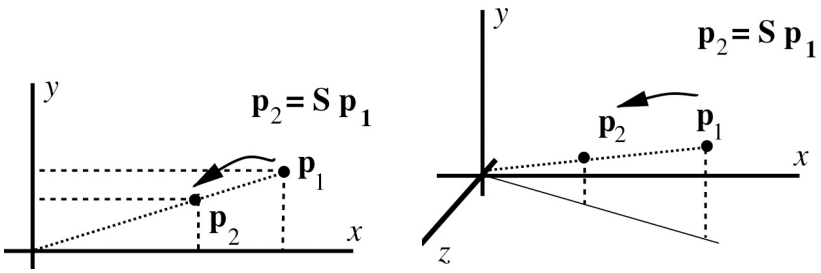
$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \left(\begin{array}{ccc|c} & \mathbf{R}_{\kappa,\phi,\omega} & & \begin{matrix}0\\0\\0\end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

Scaling of a 2D point $\mathbf{p}_2 = \mathbf{S}\mathbf{p}_1$ involves a 2×2 scale matrix $\mathbf{S} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$ so that $x_2 = s_x x_1$ and $y_2 = s_y y_1$. In homogeneous co-ordinates such a scaling is as follows:

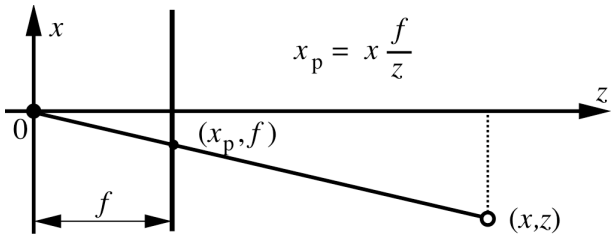
$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \left(\begin{array}{cc|c} s_x & 0 & 0 \\ 0 & s_y & 0 \\ \hline 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

Scaling of a 3D point in homogeneous coordinates is quite similar (using a 3×3 scale matrix $\mathbf{S}$:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \left(\begin{array}{ccc|c} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$



Projection onto a line or plane. An arbitrary point $\mathbf{p} = (x, z)^\mathsf{T}$ of the 2D plane $x0z$ projected along the z-axis onto the line $z - f = 0$ parallel to the x-axis produces the projected point $\mathbf{p}_\mathrm{p} = (x_\mathrm{p} = fx/z, z_\mathrm{p} = f)^\mathsf{T}$.



This projective transformation becomes linear in homogeneous coordinates:

$$\begin{pmatrix} x_\mathrm{p} \\ z_\mathrm{p} \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} xf \\ zf \\ z \end{pmatrix}$$

A 3D projection of an arbitrary point $\mathbf{p} = (x, y, z)^\mathsf{T}$ along the z-axis onto the plane $z = f$ parallel to the co-ordinate plane $x0y$ is similar to the 2D one both in Cartesian and homogeneous co-ordinates: the projected point $\mathbf{p}_\mathrm{p} = (x_\mathrm{p} = fx/z, y_\mathrm{p} = fy/z, z_\mathrm{p} = f)^\mathsf{T}$ suggests the following projective transformation in homogeneous co-ordinates:

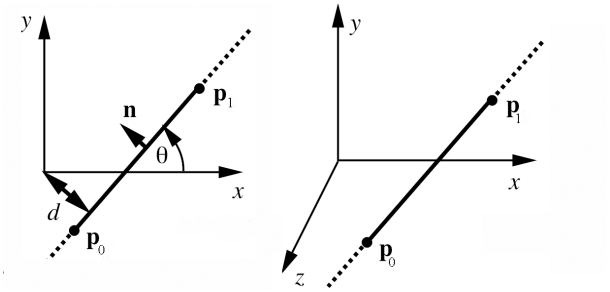
$$\begin{pmatrix} x_\mathrm{p} \\ y_\mathrm{p} \\ f \\ 1 \end{pmatrix} = \left\| \begin{array}{cccc} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{array} \right\| \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} fx \\ fy \\ fz \\ z \end{pmatrix}$$

1.2 Straight lines and segments

This section deals mostly with the Cartesian coordinates of points. Both 2D and 3D lines and straight-line segments have a very simple and convenient parametric representation: $\mathbf{p}_t = \mathbf{p}_0 + t\boldsymbol{\Delta}$ where $\mathbf{p}_0$ and $\boldsymbol{\Delta}$ are a particular starting point and a co-ordinate increment of the line, respectively, and t is a variable specifying each current position $\mathbf{p}_t$ along the line. If a segment is determined with two end points, $\mathbf{p}_0$ and $\mathbf{p}_1$, the increment is defined as $\boldsymbol{\Delta} = \mathbf{p}_1 - \mathbf{p}_0$. Points along the line $\mathbf{p}_t = \mathbf{p}_0 + t\,(\mathbf{p}_1 - \mathbf{p}_0)$ containing such a segment are easily partitioned into **interior points** such that $0 < t < 1$, **end points** with $t = 0$ and $t = 1$, and **exterior points** with $t < 0$ and $t > 1$. The same parametric representation can be rewritten as

$$\frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0} \quad \text{or} \quad \frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0} = \frac{z - z_0}{z_1 - z_0}$$

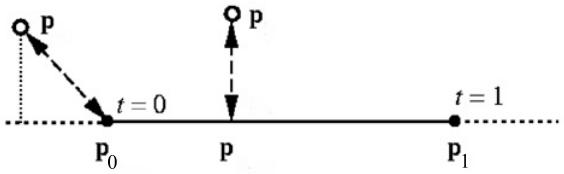
where each ratio is equal to t.



A closed-form equation for a 2D line $x\sin\theta - y\cos\theta + d = 0$ exploits a signed distance d from the co-ordinate origin $(0, 0)$ to the line and an angle θ between the line and the x-axis. In terms of the above parametric representation with $\mathbf{p}_0 = (x_0, y_0)^\mathsf{T}$ and $\boldsymbol{\Delta} = (\Delta_x, \Delta_y)^\mathsf{T}$, the closed-form parameters are as follows: $\theta = \tan^{-1}(\Delta_y/\Delta_x)$ and $d = -(x_0\sin\theta - y_0\cos\theta)$, or $d = (y_0\Delta_x - x_0\Delta_y)/\sqrt{\Delta_x^2 + \Delta_y^2}$, $\cos\theta = \Delta_x/\sqrt{\Delta_x^2 + \Delta_y^2}$, and $\sin\theta = \Delta_y/\sqrt{\Delta_x^2 + \Delta_y^2}$ The unit normal vector $\mathbf{n} = (\sin\theta, -\cos\theta)^\mathsf{T}$ is orthogonal (perpendicular) to the line, and the distance between an arbitrary point to the line is measured in the direction of the normal vector.

Distance to a segment The Cartesian distance from a given point to a 2D or 3D segment is equal to the distance to the closest point in the line if this latter point called the **projection** lies within the segment. Otherwise the distance to a segment is the distance to the closest end of this latter. When the segment is specified with its end points $\mathbf{p}_0$ and $\mathbf{p}_1$, then the position of the projection $\mathbf{p}_\mathrm{p}$ of an arbitrary point $\mathbf{p}$ is obtained by minimising the distance $d(\mathbf{p}, \mathbf{p}_t)$ from $\mathbf{p}$ to the line points $\mathbf{p}_t$, $t \in (-\infty, \infty)$:

$$t^* = \arg\min_t d(\mathbf{p}, \mathbf{p}_t) = \arg\min_t |\mathbf{p} - (\mathbf{p}_0 + t(\mathbf{p}_1 - \mathbf{p}_0)|^2$$



It is easily shown that $t^* = \Big((\mathbf{p} - \mathbf{p}_0)^{\mathrm{T}}(\mathbf{p}_1 - \mathbf{p}_0)\Big) \,/\, \Big((\mathbf{p}_1 - \mathbf{p}_0)^{\mathrm{T}}(\mathbf{p}_1 - \mathbf{p}_0)\Big)$. Thus, in the 2D case

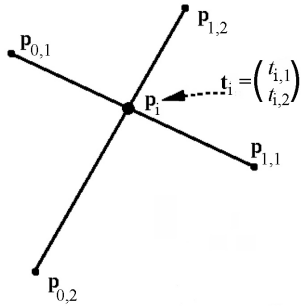
$$t^* = \frac{(x - x_0)(x_1 - x_0) + (y - y_0)(y_1 - y_0)}{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

and in the 3D case

$$t^* = \frac{(x - x_0)(x_1 - x_0) + (y - y_0)(y_1 - y_0) + (z - z_0)(z_1 - z_0)}{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

The distance from $\mathbf{p}$ to the segment with the ends $\mathbf{p}_0$ and $\mathbf{p}_1$ is equal to the inter-point distance $d(\mathbf{p}, \mathbf{p}^*)$ where $\mathbf{p}^* = \mathbf{p}_0$ if $t^* \leq 0$, $\mathbf{p}_0 + t^*(\mathbf{p}_1 - \mathbf{p}_0)$ if $0 < t^* < 1$, and $\mathbf{p}_1$ if $t^* \geq 1$.

Intersection of two 2D lines is specified with the following system of equations: $\mathbf{p}_{\mathrm{i}} = \mathbf{p}_{0,1} + t_{\mathrm{i},1}(\mathbf{p}_{1,1} - \mathbf{p}_{0,1})$ and $\mathbf{p}_{\mathrm{i}} = \mathbf{p}_{0,2} + t_{\mathrm{i},2}(\mathbf{p}_{1,2} - \mathbf{p}_{0,2})$ where $\mathbf{p}_{\mathrm{i}}$ is the intersection point. The intersection point can be interior or exterior with respect to each segment.



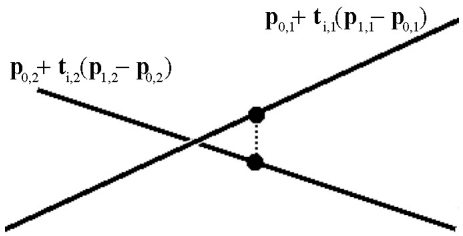
The above equation system for the intersection suggests that $\mathbf{p}_{0,1} + t_{\mathrm{i},1}(\mathbf{p}_{1,1} - \mathbf{p}_{0,1}) = \mathbf{p}_{0,2} + t_{\mathrm{i},2}(\mathbf{p}_{1,2} - \mathbf{p}_{0,2})$. If both unknown t-factors are represented as the 2×1 vector $\mathbf{t}_{\mathrm{i}} = (t_{\mathrm{i},1}, t_{\mathrm{i},2})^{\mathsf{T}}$, this relationship leads to a vector-matrix equation with 2×2 matrix and 2×1 vectors: $(\mathbf{p}_{1,1} - \mathbf{p}_{0,1}, \mathbf{p}_{0,2} - \mathbf{p}_{1,2})\,\mathbf{t}_{\mathrm{i}} = \mathbf{p}_{0,2} - \mathbf{p}_{0,1}$, or

$$\begin{pmatrix} t_{\mathrm{i},1} \\ t_{\mathrm{i},2} \end{pmatrix} = \begin{pmatrix} x_{1,1} - x_{0,1} & x_{0,2} - x_{1,2} \\ y_{1,1} - y_{0,1} & y_{0,2} - y_{1,2} \end{pmatrix}^{-1} \begin{pmatrix} x_{0,2} - x_{0,1} \\ y_{0,2} - y_{0,1} \end{pmatrix}$$

The intersection exists if the matrix is non-singular (i.e. the segments are not parallel).

Exercise: derive explicit relationships for the factors $t_{\mathrm{i},1}$ and $t_{2,2}$ specifying the the intersection point.

Intersection of two 3D lines may not exist and is approximated in that case by the closest pair of line points, one per line. These points are found by minimising the Cartesian distance $d(\mathbf{p}_{t_1}, \mathbf{p}_{t_2} = |\mathbf{p}_{t_1} - \mathbf{p}_{t_2}|^2$ between the points in both the lines $\mathbf{p}_{t_1} = \mathbf{p}_{0,1} + t_1(\mathbf{p}_{1,1} - \mathbf{p}_{0,1})$ and $\mathbf{p}_{t_2} = \mathbf{p}_{0,2} + t_2(\mathbf{p}_{1,2} - \mathbf{p}_{0,2})$.



Positioning factors $t_{\mathrm{i},1}$ and $t_{\mathrm{i},2}$ for the closest pair of the points are obtained by solving the following distance minimisation problem:

$$(t_{\mathrm{i},1}, t_{\mathrm{i},2}) = \arg\min_{t_1,t_2} |\mathbf{p}_{0,1} + t_1(\mathbf{p}_{1,1} - \mathbf{p}_{0,1}) - \mathbf{p}_{0,2} - t_2(\mathbf{p}_{1,2} - \mathbf{p}_{0,2})|^2$$

Solution of this problem (after partial derivatives of the above quadratic form by the desired factors are set equal to zero) reduces to the following linear equation

$$\begin{pmatrix} \boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_1 & -\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2 \\ -\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\Delta}_{\mathrm{dif}}^{\mathrm{T}}\boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_{\mathrm{dif}}^{\mathrm{T}}\boldsymbol{\Delta}_2 \end{pmatrix}$$

where $\boldsymbol{\Delta}_j = \mathbf{p}_{1,j} - \mathbf{p}_{0,j}$; $j = 1, 2$, and $\boldsymbol{\Delta}_{\mathrm{dif}} = \mathbf{p}_{0,1} - \mathbf{p}_{0,2}$. If the matrix is non-singular, the closest pair forming or approaching the intersection is as follows:

$$\begin{pmatrix} t_{\mathrm{i},1} \\ t_{\mathrm{i},2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_1 & -\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2 \\ -\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2 & \boldsymbol{\Delta}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 \end{pmatrix}^{-1} \begin{pmatrix} -\boldsymbol{\Delta}_{\mathrm{dif}}^{\mathrm{T}}\boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_{\mathrm{dif}}^{\mathrm{T}}\boldsymbol{\Delta}_2 \end{pmatrix}$$

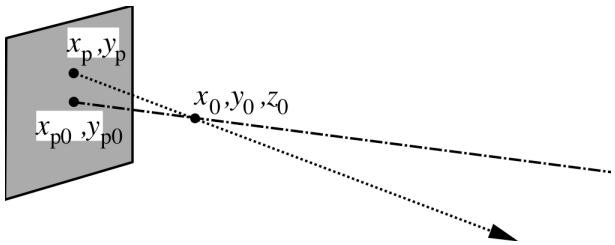
Exercise: derive explicit relationships for the factors $t_{\mathrm{i},1}$ and $t_{\mathrm{i},2}$ for the above closest pair of 3D points forming or approaching the intersection point.

1.3 Back projection of a pixel

In the simplest projection case, the optical axis coincides with the spatial Z-axis and is projected to the principal point $x_{\mathrm{pr}} = 0, y_{\mathrm{pr}} = 0$ of the image. In this case, a 3D projecting ray from a given image pixel $x_{\mathrm{pr}}, y_{\mathrm{pr}}$ back to the 3D space is easily obtained by reversing the projection, that is, by representing the 3D co-ordinates X and Y of the ray points as functions of Z. Because $X = \frac{x_{\mathrm{pr}}}{f}Z$ and $Y = \frac{y_{\mathrm{pr}}}{f}Z$ these back-projected points are

represented in homogeneous co-ordinates as follows:

$$\begin{pmatrix} X \\ Y \\ Z \\ Z \end{pmatrix} = \begin{Vmatrix} \frac{1}{f} & 0 & 0 & 0 \\ 0 & \frac{1}{f} & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{Vmatrix} \begin{pmatrix} x_{\mathrm{pr}} \\ y_{\mathrm{pr}} \\ f \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x_{\mathrm{pr}}}{f} \\ \frac{y_{\mathrm{pr}}}{f} \\ 1 \\ 1 \end{pmatrix}$$



The same back-projection scheme holds in the general case with respect to the centred image coordinates about the principal point and centred world coordinates about the optical centre (i.e. $\tilde{x}_{\mathrm{pr}} = x_{\mathrm{pr}} - x_{\mathrm{pr},0}$, $\tilde{X} = X - X_0$, etc):

$$\begin{pmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \\ \tilde{Z} \end{pmatrix} = \begin{Vmatrix} \mathbf{R}_{\kappa,\phi,\omega}^{\mathsf{T}} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{Vmatrix} \begin{Vmatrix} \frac{1}{f} & 0 & 0 & 0 \\ 0 & \frac{1}{f} & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{Vmatrix} \begin{pmatrix} \tilde{x}_{\mathrm{pr}} \\ \tilde{y}_{\mathrm{pr}} \\ f \\ 1 \end{pmatrix}$$

where the first matrix specifies the inverse rotation and scaling that reduce the centred world co-ordinate frame to the simplest projection case. Because the rotation matrix is orthogonal, its inversion is equivalent to transposition.

Let $\mathbf{P} = (q_{ij})_{i=1,2,3;j=1,\ldots,4}$ be a general-case projection matrix:

$$\begin{pmatrix} tx_{\mathrm{pr}} \\ ty_{\mathrm{pr}} \\ t \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

The optical centre $\mathbf{C} = (X_0, Y_0, Z_0, 1)^{\mathsf{T}}$ is given by the relationship $\mathbf{PC} = 0$, or

$$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} = - \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix}^{-1} \begin{pmatrix} q_{14} \\ q_{24} \\ q_{34} \end{pmatrix}$$

Coordinates of the points along every projection ray are represented as $\mathbf{C} + t\mathbf{\Delta}$ where the increment $\mathbf{\Delta} = (\Delta_x, \Delta_y, \Delta_z)^{\mathsf{T}}$ depends on the centred projected image point $(x_{\mathrm{pr}}, y_{\mathrm{pr}})$. A derivation of how the coordinates of the ray points depend on the projection parameters

is given below (a simpler derivation exploiting an infinitely far point along the projecting ray is given in Chapter 2).

Let the increment $\mathbf{\Delta}$ be normalised: $\Delta_x^2 + \Delta_y^2 + \Delta_z^2 = 1$. Then it is easily shown that

$$\begin{aligned} \tilde{x}_{\rm pr} &= \frac{q_{11}\Delta_x + q_{12}\Delta_y + q_{13}\Delta_z}{q_{11}\Delta_x + q_{12}\Delta_y + q_{13}\Delta_z} \\ \tilde{y}_{\rm pr} &= \frac{q_{21}\Delta_x + q_{22}\Delta_y + q_{23}\Delta_z}{q_{11}\Delta_x + q_{12}\Delta_y + q_{13}\Delta_z} \end{aligned}$$

Therefore,

$$\begin{aligned} (q_{11} - \tilde{x}_{\rm pr}q_{31})\Delta_x + (q_{12} - \tilde{x}_{\rm pr}q_{32})\Delta_y + (q_{13} - \tilde{x}_{\rm pr}q_{33})\Delta_z &= 0 \\ (q_{21} - \tilde{y}_{\rm pr}q_{31})\Delta_x + (q_{22} - \tilde{y}_{\rm pr}q_{32})\Delta_y + (q_{23} - \tilde{y}_{\rm pr}q_{33})\Delta_z &= 0 \\ \Delta_x^2 + \Delta_y^2 + \Delta_z^2 &= 1 \end{aligned}$$

Let $\Delta_x = \cos\phi\cos\psi$, $\Delta_y = \sin\phi\cos\psi$, and $\Delta_z = \sin\psi$ to satisfy the normalising condition. Let $\xi_{1j} = q_{1j} - \tilde{x}_{\rm pr}q_{3j}$ and $\xi_{2j} = q_{2j} - \tilde{y}_{\rm pr}q_{3j}$ for j = 1, 2, and 3. Then the above equations can be rewritten as follows:

$$\begin{aligned} \xi_{11}\cos\phi\cos\psi + \xi_{12}\sin\phi\cos\psi &= -\xi_{13}\sin\psi \\ \xi_{21}\cos\phi\cos\psi + \xi_{22}\sin\phi\cos\psi &= -\xi_{23}\sin\psi \end{aligned}$$

Let $\cos\psi = 0$. In this singular case $\xi_{13} = \xi_{23} = 0$ so that $\Delta_x = \Delta_y = 0$ and $\Delta_z = 1$. This projection ray corresponds to Z-axis ortogonal to the image plane, i.e. to the simplest, or ideal projective geometry.

Considering $\cos\phi$ and $\sin\phi$ as the components of an unknown 2×1 vector, one obtains:

$$\begin{aligned} \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} &= -\begin{pmatrix} \xi_{11} & \xi_{12} \\ \xi_{21} & \xi_{22} \end{pmatrix}^{-1} \begin{pmatrix} \xi_{13} \\ \xi_{2}3 \end{pmatrix} \tan\psi \\ &= -\frac{1}{\xi_{11}\xi_{22} - \xi_{12}\xi_{21}} \begin{pmatrix} \xi_{22} & -\xi_{12} \\ -\xi_{21} & \xi_{11} \end{pmatrix} \begin{pmatrix} \xi_{13} \\ \xi_{2}3 \end{pmatrix} \tan\psi \end{aligned}$$

Therefore,

$$\begin{aligned} \cos\phi &= \frac{\xi_{12}\xi_{23} - \xi_{13}\xi_{22}}{\xi_{11}\xi_{22} - \xi_{12}\xi_{21}} \tan\psi \\ \sin\phi &= \frac{\xi_{13}\xi_{21} - \xi_{11}\xi_{23}}{\xi_{11}\xi_{22} - \xi_{12}\xi_{21}} \tan\psi \end{aligned}$$

so that

$$\begin{aligned} \phi &= \tan^{-1}\frac{\xi_{13}\xi_{21} - \xi_{11}\xi_{23}}{\xi_{12}\xi_{23} - \xi_{13}\xi_{22}} \\ \psi &= \tan^{-1}\frac{\xi_{11}\xi_{22} - \xi_{12}\xi_{21}}{\sqrt{(\xi_{12}\xi_{23} - \xi_{13}\xi_{22})^2 + (\xi_{13}\xi_{21} - \xi_{11}\xi_{23})^2}} \end{aligned}$$

Using these values, one can easily find the desired increments:

$$\Delta_x = \cos\phi\cos\psi = \frac{\xi_{12}\xi_{23} - \xi_{13}\xi_{22}}{\sqrt{(\xi_{12}\xi_{23} - \xi_{13}\xi_{22})^2 + (\xi_{13}\xi_{21} - \xi_{11}\xi_{23})^2 + (\xi_{11}\xi_{22} - \xi_{12}\xi_{21})^2}}$$

$$\Delta_y = \sin\phi\cos\psi = \frac{\xi_{13}\xi_{21} - \xi_{11}\xi_{23}}{\sqrt{(\xi_{12}\xi_{23} - \xi_{13}\xi_{22})^2 + (\xi_{13}\xi_{21} - \xi_{11}\xi_{23})^2 + (\xi_{11}\xi_{22} - \xi_{12}\xi_{21})^2}}$$

$$\Delta_z = \sin\psi = \frac{\xi_{11}\xi_{22} - \xi_{12}\xi_{21}}{\sqrt{(\xi_{12}\xi_{23} - \xi_{13}\xi_{22})^2 + (\xi_{13}\xi_{21} - \xi_{11}\xi_{23})^2 + (\xi_{11}\xi_{22} - \xi_{12}\xi_{21})^2}}$$

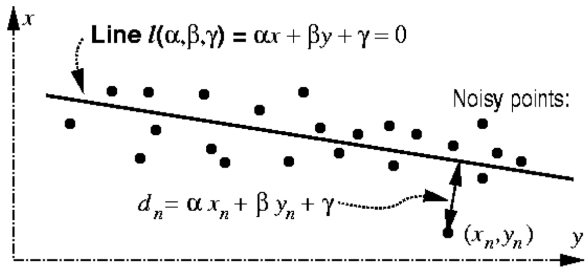
1.4 Least squares line fitting

The least squares criterion is to minimise the total or mean square distance from a given set of points to the line $l(\alpha, \beta, \gamma) = \alpha x + \beta y + \gamma$ such that $\alpha^2 + \beta^2 = 1$:

$$(\alpha^*, \beta^*, \gamma^*) = \arg \min_{\alpha,\beta,\gamma \,:\, \alpha^2+\beta^2=1} D(\alpha, \beta, \gamma)$$

where $D(\alpha, \beta, \gamma)$ denotes the total distance that accumulates the individual distances $d_n^2 = (\alpha x_n + \beta y_n + \gamma)^2$ from the points $\mathbf{p}_n = (x_n, y_n)$ to the line:

$$= \sum_{n=1}^{N} d_n^2 \equiv \sum_{n=1}^{N} (\alpha x_n + \beta y_n + \gamma)^2$$



Unconstrained minimisation of $D(\alpha, \beta, \gamma)$ by γ is performed by setting to zero the corresponding partial derivative:

$$\frac{\partial D(\alpha, \beta, \gamma)}{\partial \gamma} = 0 \rightarrow \frac{\partial}{\partial \gamma} \sum_{n=1}^{N} (\alpha x_n + \beta y_n + \gamma)^2 = 2 \sum_{n=1}^{N} (\alpha x_n + \beta y_n + \gamma) = 0$$

so that the optimum $\gamma^* = -(\alpha \bar{x} + \beta \bar{y})$ where $\bar{x}$ and $\bar{y}$ are the mean coordinates:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n; \quad \bar{x} = \frac{1}{N} \sum_{n=1}^{N} y_n$$

After substituting the γ^* value, the constrained Lagrange minimisation of $D(\alpha, \beta, \gamma^*)$ by α and β is performed as follows: $(\alpha^*, \beta^*) = \arg\min_{\alpha,\beta} \Phi(\alpha, \beta)$ where

$$\Phi(\alpha, \beta) = \sum_{n=1}^{N} [\alpha(x_n - \bar{x}) + \beta(y_n - \bar{y})]^2 - \lambda(\alpha^2 + \beta^2 - 1)$$

and λ is the Lagrange factor. By setting to zero the partial derivatives of $\Phi(\alpha, \beta)$:

$$\begin{aligned} \frac{\partial}{\partial\alpha}\Phi(\alpha,\beta) &= 2\sum_{n=1}^{N}\left[\alpha(x_n-\bar{x})+\beta(y_n-\bar{y})\right](x_n-\bar{x}) - 2\lambda\alpha = 0 \\ \frac{\partial}{\partial\beta}\Phi(\alpha,\beta) &= 2\sum_{n=1}^{N}\left[\alpha(x_n-\bar{x})+\beta(y_n-\bar{y})\right](y_n-\bar{y}) - 2\lambda\beta = 0 \end{aligned}$$

the minimisation is reduced to the following eigen-vector problem:

$$\begin{pmatrix} \mu_{xx} & \mu_{xy} \\ \mu_{xy} & \mu_{yy} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} - \lambda \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0 \text{ or } \begin{pmatrix} \mu_{xx} - \lambda & \mu_{xy} \\ \mu_{xy} & \mu_{yy} - \lambda \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0 \quad (1.1)$$

where $\mu_{xx} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x})^2$ and $\mu_{yy} = \frac{1}{N-1}\sum_{n=1}^{N}(y_n - \bar{y})^2$ are the variances of the x- and y-coordinates, respectively, and $\mu_{xy} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})$ is the covariance of the coordinates of the points to be approximated with the line. To solve the minimisation problem, the eigen-vector corresponding to the smallest eigen-value has to be computed. In this 2×2 case it is easily found analytically:

$$\begin{pmatrix} \alpha^* \\ \beta^* \end{pmatrix} = \frac{1}{\sqrt{\mu_{xy}^2 + (\lambda^* - \mu_{xx})^2}} \begin{pmatrix} \mu_{xy} \\ \lambda^* - \mu_{xx} \end{pmatrix}$$

where $\lambda^* = \frac{1}{2}\left(\mu_{xx} + \mu_{yy} - \sqrt{(\mu_{xx} - \mu_{yy})^2 + 4\mu_{xy}}\right)$.

Exercise: Obtain the above relationship from the initial equation 1.1.

1.5 Line detection with Hough transform

Each straight line $l(x, y|\theta^\circ, p^\circ) = x\cos\theta^\circ + y\sin\theta^\circ - p = 0$ is represented by the point (θ°, p°) in the 2D space (or plane) of parameters (θ, p). Each point $(x_\mathsf{A}, y_\mathsf{A}$ on that line in principle may belong to an infinite subset of other straight lines. As shown in Fig. 1.1, all the lines passing through a point $(x_\mathsf{A}, y_\mathsf{A})$ relate to a wave-like trajectory on the parameter plane:

$$h(\theta, p|x_\mathsf{A}, y_\mathsf{A}) = x_\mathsf{A}\cos\theta + y_\mathsf{A}\sin\theta - p = 0$$

Ideally, the trajectories for the points of the same line $l(x, y|\theta^\circ, p^\circ)$ intersect in the same point θ°, p° of the parameter space (see Fig. 1.2). In practice, the intersections may

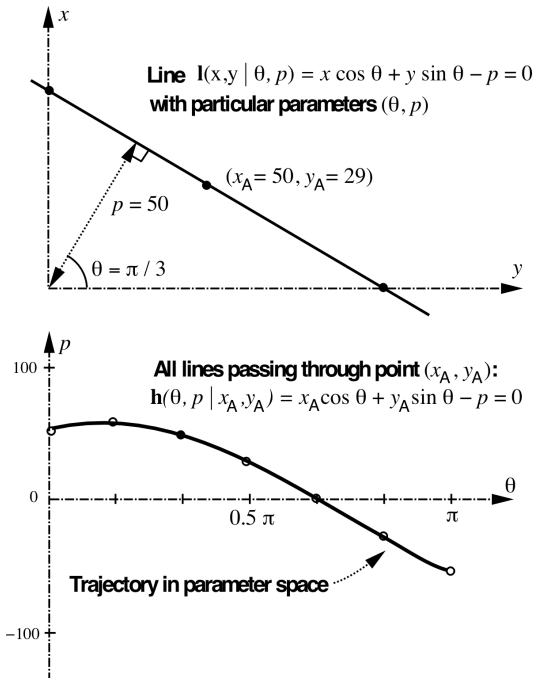


Figure 1.1: Hough transform.

be inexact due to noisy positions of the points in a digital image and of their trajectories. A proper quantisation of the parameter space accounts for possible inexact intersections. Then the search for the intersection point can be replaced with the search for a spatial cluster of the points along all the trajectories for a given set $\mathbf{p}_n = (x_n, y_n); n = 1, \ldots, N$, of points.

Algorithm of line detection in an image g with the Hough transform assuming that their candidate points are edge ones with relatively large signal gradients is as follows:

1. Quantise the parameter space between appropriate maximum and minimum values p and θ, e.g. $\theta \in [0, \pi]$ and $p \in [0, p_{\max} = \sqrt{x_{\max}^2 + y_{\max}^2}]$.

2. Given the quantisation steps δ_θ and δ_p, form an accumulator array $[A(i,j)]_{i,j=0,0}^{I,J}$ of size IJ where $I = \pi/\delta_\theta$ and $J = p_{\max}/\delta_p$, whose elements are initially set to zero: $A(i,j) = 0$.

3. For each edge point (x, y) in the image g, that is, the point where the gradient magnitude exceeds a given threshold, increment all the accumulator elements $A(p, \theta)$ along the corresponding curvilinear trajectory, i.e. $A(p, \theta) \leftarrow A(p, \theta) + 1$ for p and θ satisfying the trajectory equation $x \cos\theta + y \sin\theta - p = 0$ to within the precision of parameter quantisation.

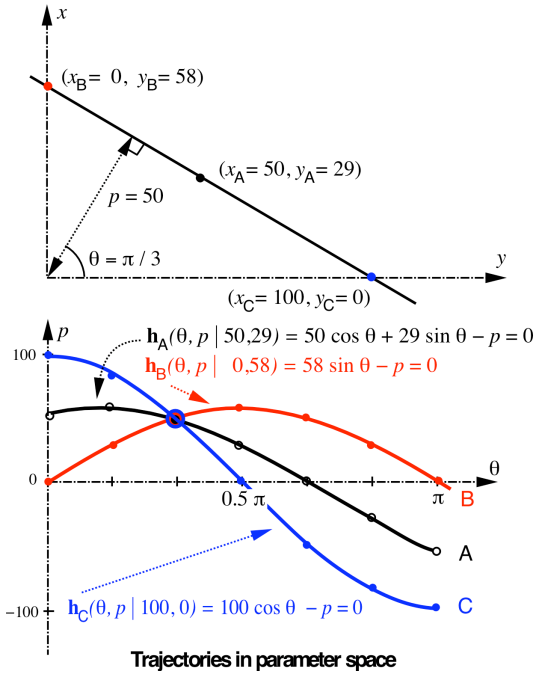


Figure 1.2: Hough transform: intersecting trajectories.

4. Find local maxima in the accumulator array that now correspond to collinear edge points in the image (usually, the accumulator A is first smoothed, and then peaks are determined as values greater than a threshold).

5. After the peaks are obtained, detemine the parameter pairs for each line as the centre points of the peaks and find the line corresponding to each determined parameter pair (θ, p) by examining the edge points near the line in the image (see Fig. 1.3).

The above Hough transform can be simplified by using edge directions because the trajectory dimension in the parameter space is reduced when the direction is also assigned to the edge points by an edge operator. The equation of the straight line passing through the point $(x_\mathsf{A}, y_\mathsf{A})$ along the direction θ_A is uniquely determined as $x\cos\theta_\mathsf{A} + y\sin\theta_\mathsf{A} - (x_\mathsf{A}\cos\theta_\mathsf{A} + y_\mathsf{A}\sin\theta_\mathsf{A}) = 0$. Therefore, the line is transformed to the single point $(\theta, x_\mathsf{A}\cos\theta_\mathsf{A} + y_\mathsf{A}\sin\theta_\mathsf{A})$ in the parameter space of the Hough transform. The calculation of the accumulator array in this case is very simple, and the peaks are much easier found by analysing point clusters exemplified by Fig. 1.4.

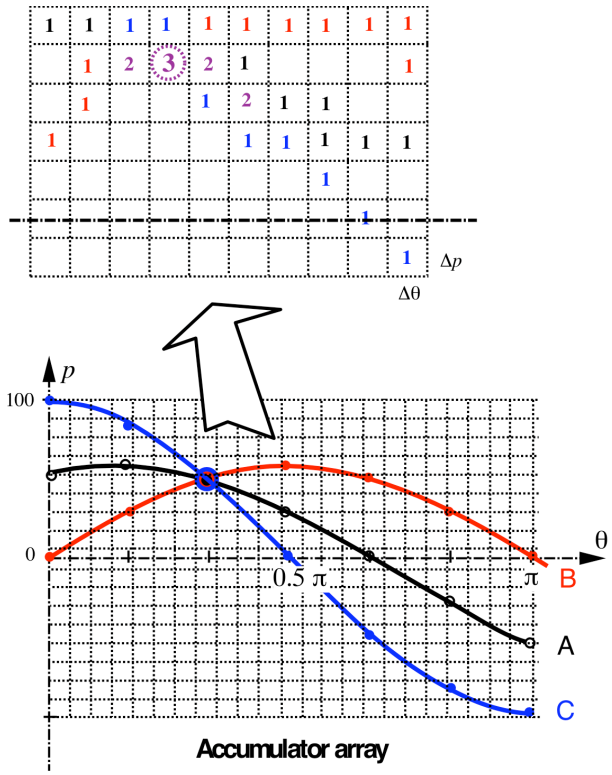


Figure 1.3: Implementation of the Hough transform.

Figure 1.4: The edge-based Hough transform.

Chapter 2

Camera Calibration

2.1 Pin-hole camera model

The two key questions in understanding image formation for a 3D scene are where some point of the scene appears in the image and what determines its brightness or colour. The first question is answered with a geometric camera model that gives a precise mathematical description of geometric relationships between the real 3D world and the images perceived. Most typically the 3D scenes are projected onto the 2D images by projective perspective transformations. The second question involves a radiometric model that takes account of general illumination and optical properties of visible surfaces having both diffuse and specular reflection components. Different points on an object in front of the visual sensor produce different values in the pixels, depending on the amount of incident radiance, how they are illuminated, how they reflect light, how the reflected light is collected by a lens system, and how the sensor responds to the incoming light (Fig. 2.1).

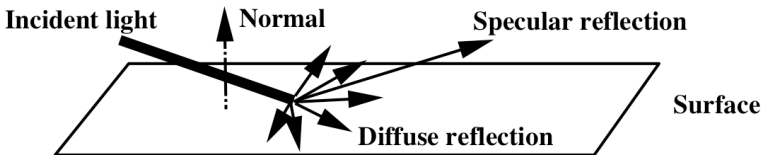


Figure 2.1: Light reflection.

Pin-hole camera geometry. Let us consider the box in Fig. 2.2 with a small hole punched in the front face. Some of the light rays, emitted or reflected by 3D objects, pass through the hole and form an inverted image of these objects on the back face. The operation creating the inverted image is called a **perspective projection**. Such an ideal pinhole projection is the simplest geometric camera model which also holds for fixed-focus cameras having general-purpose lens systems. As underscored in [4], the pinhole

camera accurately models the geometry and optics of most of the modern cameras. The pinhole is the camera's optical, or **focal centre** with the coordinates $\mathbf{O} = [X_0, Y_0, Z_0]$ in a certain real-world co-ordinate system $0XYZ$. The optical centre can be also denoted $\mathbf{C}$ below.

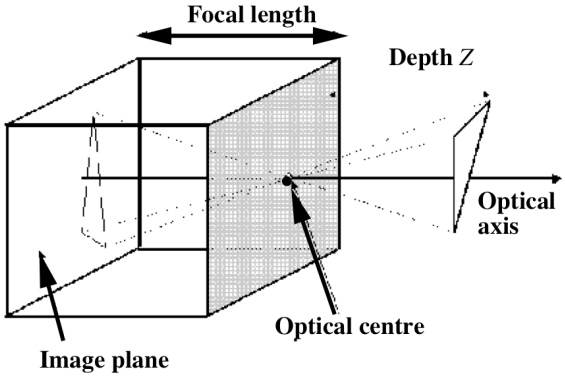


Figure 2.2: Pin-hole camera.

The image is formed in the **image, or retinal plane** through a perspective projection. The camera's **optical axis** crossing the optical centre is perpendicular to the image plane. The distance from the optical centre to the image plane along the optical axis is the camera's **focal length**, or focal distance, or camera constant f. The plane that contains the optical centre and is parallel to the image plane is called the focal plane.

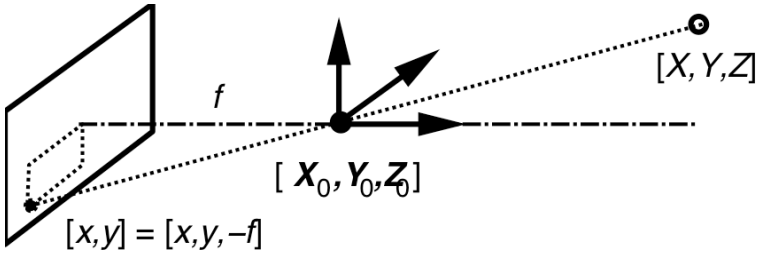


Figure 2.3: Pin-hole projection.

Every 3D point $\mathbf{S} = [X, Y, Z]$ in the field of view is projected onto a corresponding point $\mathbf{s} = [x, y]$ on the image plane. Let a 3D point **S** being at the depth Z from the optical centre be projected through this centre (Fig. 2.3). Then it is projected into the point that is a trace of the line **OS** in the image plane **R**. Geometrically, a projection of an object onto a plane placed at the focal distance in front of the focal plane is similar to the rectified projection obtained in the image plane as shown in Fig. 2.4. Traditionally, the image plane is drawn in front of the focal point.

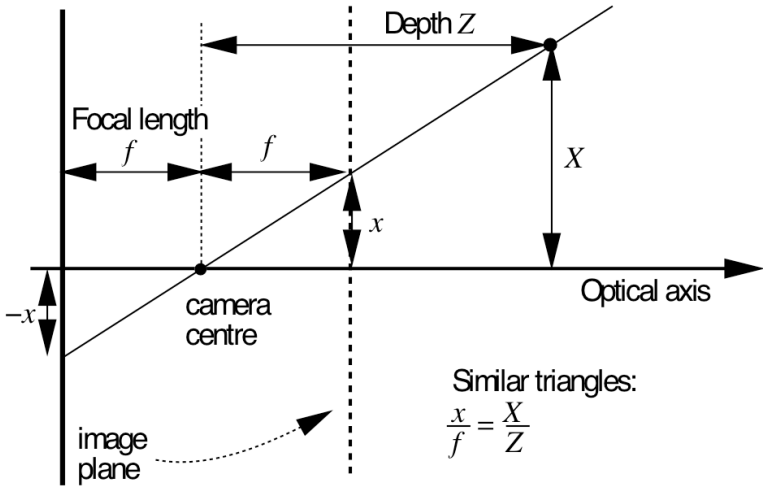


Figure 2.4: Pin-hole camera model.

Camera reference frame. There might be an infinite variety of 3D world reference frames (WRF) giving Cartesian co-ordinates of spatial points with respect to different origins and orientations of coordinate axes and in different linear units (μm, mm, inches, light years, parsecs, etc). In the image reference frame, 2D image points are typically measured in pixels. Each pixel (or picture element) is of the size of a camera's photoreceptor (e.g. about 15×15 μm in a conventional CCD camera). Generally, a geometric transformation that maps Cartesian point coordinates in one reference frame onto coordinates in another frame involves an Euclidean motion (translation and rotation) and scaling (see Fig. 2.5). In the simplest case, the scale of the coordinates is the same in the both reference frames, and the transformation involves only rotation and translation. The corresponding transformation matrix maps a 3D point in the WRF, $(x, y, z, 1)^{\mathsf{T}}$ in homogeneous coordinates, into its counterpart, $(x', y', z', 1)^{\mathsf{T}}$ in the camera reference frame (CRF). This 4×4 matrix combining rotation and translation in homogeneous co-ordinates is written as:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Here, the matrices $\mathbf{R}$ and $\mathbf{T}$ describe, respectively, the orientation and position of the CRF relative to the WRF. These characteristics are called *extrinsic* camera parameters.

The camera model has the simplest form when the image and spatial coordinates are

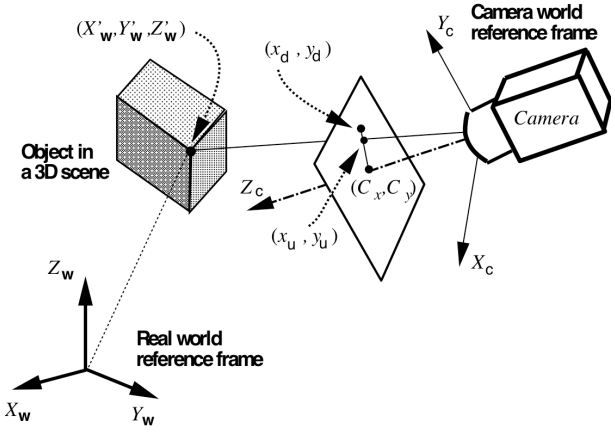


Figure 2.5: World and image reference frames.

given in the CRF:

$$\begin{bmatrix} tx \\ ty \\ t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} TX \\ TY \\ TZ \\ T \end{bmatrix}, \quad \text{or} \quad x = f\frac{X}{Z};\ y = f\frac{Y}{Z}$$

Generally, we need to determine the CRF with respect to a particular WRF, given the camera coordinates $\mathbf{s} = [x, y]^\mathsf{T}$ of the observed perspective projection points and the world coordinates $\mathbf{S}_\mathrm{w} = [X_\mathrm{w}, Y_\mathrm{w}, Z_\mathrm{w}]^\mathsf{T}$ of the corresponding 3D object points.

Let lens distortions can be ignored. Let the unit vectors $\mathbf{X}_\mathrm{c}$, $\mathbf{Y}_\mathrm{c}$, $\mathbf{Z}_\mathrm{c}$ define the X, Y, or Z-axis of the camera reference frame, respectively, in the world coordinates. Let (x_0, y_0) be the coordinates of the principal point in the image assuming that the origin does not coincide with the trace of the optical axis. Then the camera model with respect to the world reference frame is as follows:

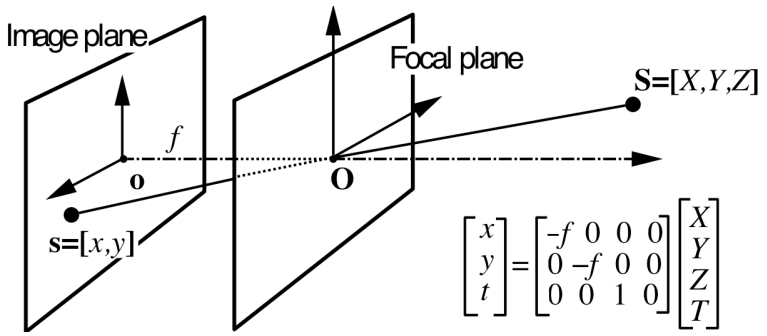
$$\frac{x - x_0}{f} = \frac{(\mathbf{S}_\mathrm{w} - \mathbf{O}) \bullet \mathbf{X_c}}{(\mathbf{S}_\mathrm{w} - \mathbf{O}) \bullet \mathbf{Z_c}};\ \frac{y - y_0}{f} = \frac{(\mathbf{S}_\mathrm{w} - \mathbf{O}) \bullet \mathbf{Y_c}}{(\mathbf{S}_\mathrm{w} - \mathbf{O}) \bullet \mathbf{Z_c}}$$

where $\bullet$ denotes the dot product of vectors. The above dot products give the co-ordinates X, Y, or Z in the CRF for the point $\mathbf{S}$ projected onto the image point $\mathbf{s} = [x, y]$.

2.2 More on image and focal planes

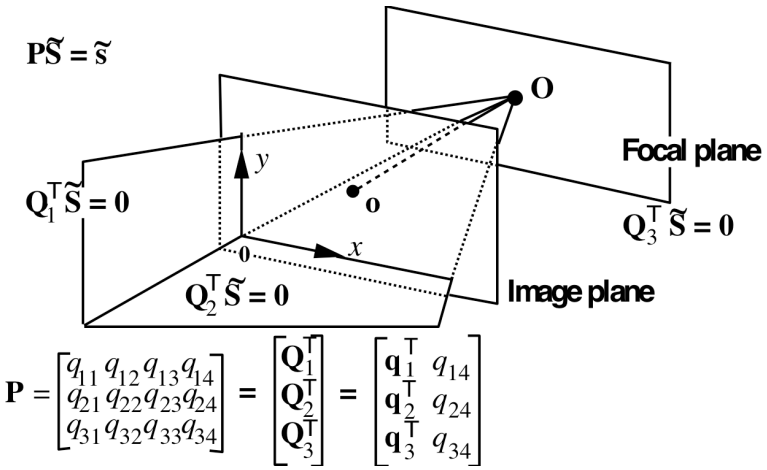
As outlined above, an optical image is formed by a perspective projection of an optical 3D surface onto the image plane of a sensor via an optical centre $\mathbf{O}$ located at the focal distance f of the optical system. The projection $\mathbf{s} = [x, y]^\mathsf{T}$ of a 3D point $\mathbf{S} = [X, Y, Z]^\mathsf{T}$

is formed by intersecting the projecting ray $\overline{\mathbf{SO}}$ with the image plane. The focal plane contains the centre $\mathbf{O}$ and is parallel to the image plane.



Perspective projection is a relationship between the image and 3D spatial coordinates in the standard CRF (coordinate system of the camera):

$$-\frac{f}{Z} = \frac{x}{X} = \frac{y}{Y}.$$



The standard CRF has the X- and Y-axes parallel to the x- and y-axes of the image, respectively. The Z-axis coincides with the optical axis of the camera that is perpendicular to the image plane and goes through the optical centre.

In an arbitrary WRF, the perspective projection in homogeneous coordinates $\tilde{\mathbf{S}} = [X, Y, Z, T]^{\mathrm{T}}$ and $\tilde{\mathbf{s}} = [x, y.t]^{\mathrm{T}}$ is as follows: $\mathbf{P}\tilde{\mathbf{S}} = \tilde{\mathbf{s}}$ where $\mathbf{P}$ is a particular 3×4 projection matrix. The matrix has a simple geometric interpretation. Its row vectors $\mathbf{q}_{i^{\mathsf{T}}}$; $i = 1, \ldots, 3$, specify the projective planes with the point equation $\mathbf{q}_i^{\mathsf{T}}\mathbf{S} = 0$ corresponding to points in the image plane with $x = 0$, $y = 0$, and $t = 0$, respectively.

The plane of the equation $\mathbf{q}_3^{\mathsf{T}}\mathbf{S} = 0$ yielding $t = 0$ corresponds to image points at infinity, that is, it represents the focal plane. The intersection $\mathbf{q}_i^{\mathsf{T}}\mathbf{S} = 0$; $i = 1, 2$ of the two other planes is the line going through the optical centre $\mathbf{O}$ and the coordinate origin $\mathbf{o}$ in the image plane. The optical centre $\mathbf{O}$ is defined as the intersection of the three planes $\mathbf{q}_i^{\mathsf{T}}\mathbf{S} = 0$; $i = 1, 2, 3$, and can be obtained by solving the system of three linear equations.

Optical centre and projecting ray. The projection matrix can be rewritten as $\mathbf{P} = [\mathbf{Q} \;\; \mathbf{q}]$ where

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}; \quad \mathbf{q} = \begin{bmatrix} q_{14} \\ q_{24} \\ q_{34} \end{bmatrix}$$

If the matrix $\mathbf{Q}$ is of rank 3 then the optical centre is obtained by solving the linear system $\mathbf{P}\tilde{\mathbf{O}} = \mathbf{0}$, that is,

$$\mathbf{O} = -\mathbf{Q}^{-1}\mathbf{q}.$$

The *optical ray* defined by a pixel $\mathbf{s}$ is going through the optical centre $\mathbf{O}$ and the point at infinity, $\tilde{\mathbf{D}}$, with homogeneous (or projective) coordinates $[\mathbf{D}^\mathsf{T}, 0]^\mathsf{T}$ that satisfies equation $\mathbf{s} = \mathbf{P}\mathbf{D}$. Therefore, $\mathbf{D} = \mathbf{Q}^{-1}\mathbf{s}$, and a point on the ray is given by

$$\mathbf{Q}^{-1}(-\mathbf{q} + \lambda\mathbf{s}) = \mathbf{O} + \lambda\mathbf{Q}^{-1}\mathbf{s}$$

where $-\infty < \lambda < \infty$.

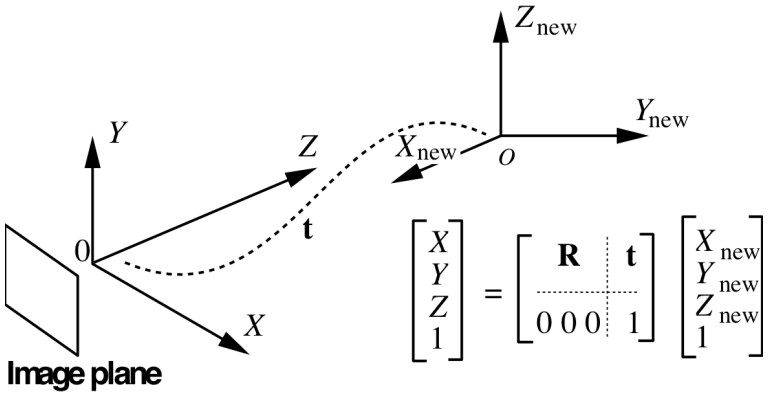
Optical and digital coordinates In the standard camera geometry, the origin of the image coordinates coincides with the **principal point**, or the trace (intersection point) of the optical axis serving as the axis Z, and the 3D and 2D point coordinates X, Y, Z, x, y and the focal length f are given in the length units (e.g., millimeters). **Intrinsic** camera parameters for sensing digital images include the origin, $(c_{x,0}, c_{y,0})$, of the pixel positions with respect to the principal point, where c_x and c_y denote the row and column pixel position, respectively, and the scale factors, $k_{c,x}$ and $k_{c,y}$, expressed in the relative units (pixel / meter) and inversely proportional to the horizontal and vertical size of the pixel. The intrinsic camera parameters $f, k_{c,x}, k_{c,y}, c_{x,0}, c_{y,0}$ specify digital image coordinates, that is, pixel positions in a digital image, as follows:

$$c_x = k_{c,x}x + c_{x,0} = -fk_{c,x}\frac{X}{Z} + c_{x,0}; \quad c_y = k_{c,y}y + c_{y,0} = -fk_{c,y}\frac{Y}{Z} + c_{y,0}.$$

Thus the projection equation for a digital image (in terms of pixels) is $\mathbf{s}_{\text{pix}} = \mathbf{K}\mathbf{s}$:

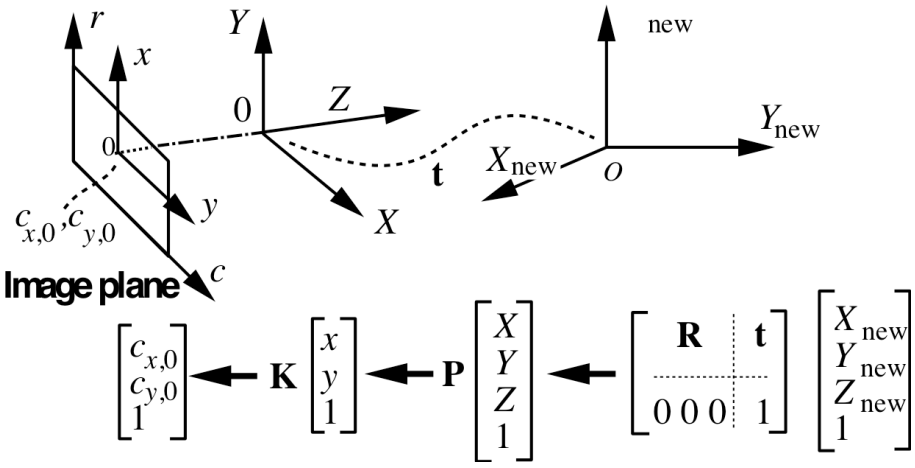
$$\begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} = \begin{bmatrix} k_{c,x} & 0 & c_{x,0} \\ 0 & k_{c,y} & c_{y,0} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_{c,x} & 0 & c_{x,0} & 0 \\ 0 & -fk_{c,y} & c_{y,0} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

An Euclidean transformation of the 3D space consisting in rotation of the old co-ordinate system followed by translation changes the co-ordinates of the points.



The matrix $\mathbf{R}$ and the vector $\mathbf{t}$ describing the orientation and positioning of the camera with respect to the new WCF (world co-ordinate frame) are called **extrinsic** camera parameters.

General form of the matrix $\mathbf{P}$ is as follows.



By combining co-ordinate transformations and projection in the camera coordinate system, one obtains a general perspective projection matrix for projecting a 3D point in the world coordinates onto the digital image:

$$\mathbf{P}_{\text{new}} = \begin{bmatrix} k_{c,x} & 0 & c_{x,0} \\ 0 & k_{c,y} & c_{y,0} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & t_x \\ \mathbf{r}_2 & t_y \\ \mathbf{r}_3 & t_z \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{r}_i$ is the i-th row of the rotation matrix $\mathbf{R}$ specifying the camera orientation.

2.3 Camera calibration

Camera calibration is a process of determining parameters of the imaging process, that is, of the projective transformation that maps a 3D point in a given WRF, onto its 2D image with co-ordinates measured in pixel units. Such a transformation, when determined, relates the image measurements to the spatial structure of the observed scene. The calibration determines both the external, or extrinsic, parameters relating the WRF to the CRF and internal, or intrinsic, parameters of the camera. Rays back-projected from image pixels into the 3D space have the following properties: (*i*) an angle between two rays relates to respective positions of two pixels, (*ii*) a ray and a depth value provide the position of a 3D point, and (*iii*) rays from two cameras viewing the same 3D scene intersect at a 3D position of each point depicted in the both images. Let a 3D point in the standard CRF of a pinhole camera model be projected into the image plane placed at the focal distance f from the origin of the CRF (optical centre). Let the centre of image 2D co-ordinates be in the principal point, or trace of the optical axis. Then in homogeneous image co-ordinates a perspective projection is given by the following 4×4 matrix (see Fig. 2.6):

$$\begin{pmatrix} sx_{\mathrm{p}} \\ sy_{\mathrm{p}} \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}; \quad s \neq 0$$

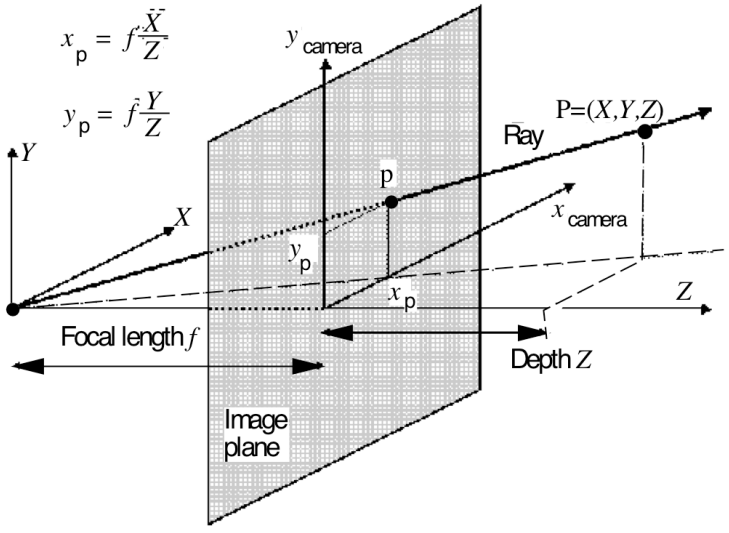


Figure 2.6: Perspective projection.

Standard projective geometry and image distortions. In image processing, the origin of the image reference frame is usually at the top left of the image, x-axis and y-axis pointing rightward and downward, respectively. Image coordinates are measured in pixels, and their metric counterparts can be obtained by accounting for the size of photosensors transforming the light rays into electrical signals. As shown in Fig. 2.7, the sensors are not necessarily square. Let d_{x} and d_{y} be the x- and y-size of the rectangular sensor, that is, the center-to-center horizontal or vertical distance between the adjacent sensing elements, respectively. Then the coordinates of an image point in the pixel-based image reference frame are as follows:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} -{d_x}^{-1} & 0 & x_c \\ 0 & -{d_y}^{-1} & y_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\mathrm{c}} \\ y_{\mathrm{c}} \\ 1 \end{pmatrix}$$

The values d_x and d_y are usually provided by the camera manufacturers. d_y should be doubled if only even or odd fields are used instead of the full size frames.

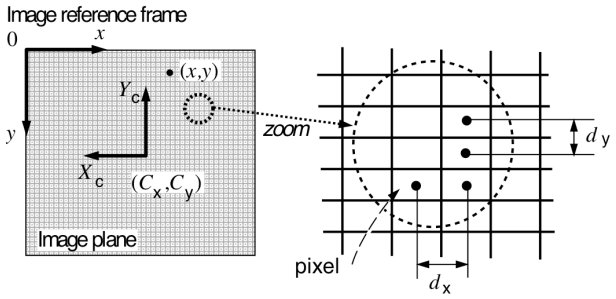


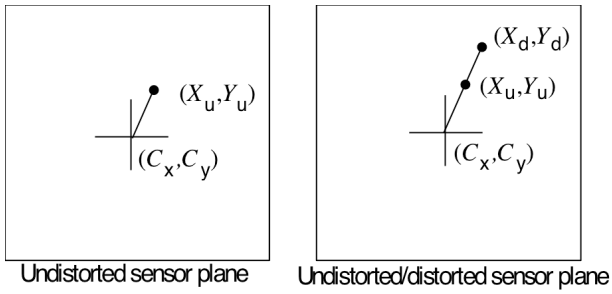
Figure 2.7: Pixel coordinates.

Today's CCD cameras have different sizes, numbers, sampling rates, and other characteristics of sensing elements. The camera parameters include also the numbers $N_{\mathrm{c},x}$ and $N_{\mathrm{c},y}$ of the sensing elements in the horizontal and vertical direction that specify the image size (the total number of the sensed pixels) as well as the number $N_{\mathrm{f},x}$ of pixels in a line sampled by the computer. A more general representation of the coordinates of an image point in the pixel-based image reference frame is:

$$X_{\mathrm{f}} = s_{\mathrm{x}} \frac{X_{\mathrm{d}}}{d'_{\mathrm{x}}} + C_{\mathrm{x}}; \;\; Y_{\mathrm{f}} = \frac{Y_{\mathrm{d}}}{d_{\mathrm{y}}} + C_{\mathrm{y}}$$

where $d'_{\mathrm{x}} = d_{\mathrm{x}} \frac{N_{\mathrm{c},x}}{N_{\mathrm{f},x}}$ and s_{x} is a scale factor accounting for uncertainty due to a framegrabber horizontal scanline resampling and acquisition timing error. A 1% error in characteristics of the sensing elements may yield errors up to 3 to 5 pixels in a full frame image [13].

Most of the existing calibration methods assume that the optical and geometrical camera design follows the pinhole camera model. Possible deviations of actual camera characteristics from the pinhole model are taken into account in some calibration methods by using additional parameteric models of image distortions:



Standard calibration procedures account for only a first order radial lens distortion described by non-linear model with a single distorting parameter, κ_1. The coefficient κ_1 relates the distorted (true) and undistorted (ideal) image coordinates as follows:

$$X_d(1 + \kappa_1({X^2}_d + {Y^2}_d)) = X_u$$
$$Y_d(1 + \kappa_1({X^2}_d + {Y^2}_d)) = Y_u$$

Projective geometry: general case. Generally, the projection matrix $\mathbf{P}$ depends on both the intrinsic and extrinsic camera parameters:

$$\mathbf{P}_{\text{new}} = \begin{bmatrix} -k_{c,x}f\mathbf{r}_1 + c_{x,0}\mathbf{r}_3 & -k_{c,x}ft_x + c_{x,0}t_z \\ -k_{c,y}f\mathbf{r}_2 + c_{y,0}\mathbf{r}_3 & -k_{c,y}ft_y + c_{y,0}t_z \\ \mathbf{r}_3 & t_z \end{bmatrix}$$

where the vectors $\mathbf{r}_i$ are the row vectors of the rotation matrix $\mathbf{R}$. If $k_{c,x}$ and $k_{c,y}$ are not zero, the matrix $\mathbf{P}_{\text{new}}$ is of rank 3.

To estimate the intrinsic and extrinsic parameters of a camera, one has first to estimate the projection matrix $\mathbf{P} \equiv \mathbf{P}_{\text{new}}$. Then, if necessary, the camera parameters should be derived from $\mathbf{P}$. Special constraints on the projection matrix $\mathbf{P}$ hold.

Theorem 1 (O. Faugeras) Let $\mathbf{P}$ be a 3×4 projection matrix of rank 3:

$$\mathbf{P} = \begin{bmatrix} \mathbf{q}_1^{\mathsf{T}} & q_{14} \\ \mathbf{q}_2^{\mathsf{T}} & q_{24} \\ \mathbf{q}_3^{\mathsf{T}} & q_{34} \end{bmatrix}.$$

There exist four sets of extrinsic and intrinsic parameters such that $\mathbf{P}$ can be written as

$$\mathbf{P} = \begin{bmatrix} -k_{c,x}f\mathbf{r}_1 + c_{x,0}\mathbf{r}_3 & -k_{c,x}ft_x + c_{x,0}t_z \\ -k_{c,y}f\mathbf{r}_2 + c_{y,0}\mathbf{r}_3 & -k_{c,y}ft_y + c_{y,0}t_z \\ \mathbf{r}_3 & t_z \end{bmatrix}$$

if and only if the two constraints are satisfied: $||\mathbf{q}_3|| = 1$ *and* $(\mathbf{q}_1 \times \mathbf{q}_3) \bullet (\mathbf{q}_2 \times \mathbf{q}_3) = 0$.

Here, $||\ldots||$, $\bullet$ and $\times$ denote the vector norm, the vector dot product, and the vector cross product, respectively. Geometric interpretation of the theorem is as follows. The direction of the line $c_x = 0$ or $c_y = 0$ in the image plane is that of the intersection of the focal plane $\mathbf{q}_3^\mathsf{T}\mathbf{S} = 0$ with the plane $\mathbf{q}_1^\mathsf{T}\mathbf{S} = 0$ or $\mathbf{q}_2^\mathsf{T}\mathbf{S} = 0$, respectively. This direction is given by the vector cross product $\mathbf{q}_1 \times \mathbf{q}_3$ or $\mathbf{q}_2 \times \mathbf{q}_3$, respectively. The dot product $(\mathbf{q}_1 \times \mathbf{q}_3) \bullet (\mathbf{q}_2 \times \mathbf{q}_3)$ specifies the cosine of the angle (defined modulo π) between those two lines. The image plane can be placed either behind or in front of the optical centre.

Number of parameters. If physically the sensor axes $c_x = 0$ and $c_y = 0$ are orthogonal, there are the two constraints for the projection matrix, and 10 intrinsic and extrinsic camera parameters have to be estimated. If the axes may have an arbitrary angle, θ, there is only one constraint $||\mathbf{q}_3|| = 1$, and 11 camera parameters, including the intrinsic angle θ, have to be estimated. There are only four equivalent variants of the estimates differing in that the origin of coordinates is in front of the camera ($t_z > 0$) or behind it ($t_z < 0$) or the axes $c_x = 0$ and $c_y = 0$ are direct or inverted in the image plane.

Linear calibration methods. Calibration data contain N 3D reference points $\mathbf{S}_i = [X_i, Y_i, Z_i]$ and corresponding 2D image coordinates $\mathbf{s}_i = [c_{x,i}, c_{y,i}]$. The following linear relationship between the 3D and 2D coordinates holds:

$$\mathbf{q}_1^\mathsf{T}\mathbf{S}_i - c_{x,i}\mathbf{q}_3^\mathsf{T}\mathbf{S}_i + q_{14} - c_{x,i}q_{34} = 0; \quad \mathbf{q}_2^\mathsf{T}\mathbf{S}_i - c_{y,i}\mathbf{q}_3^\mathsf{T}\mathbf{S}_i + q_{24} - c_{y,i}q_{34} = 0$$

Every reference point $\mathbf{S}_i$ gives two linear equations in the unknowns $\mathbf{q}_m^\mathsf{T}$ and q_{m4}; $m = 1, 2, 3$, and the N reference points result in an over-determined system of $2N$ homogeneous linear equations $\mathbf{A}\mathbf{q} = 0$. The latter can be solved by the least-square method: $\min ||\mathbf{A}\mathbf{q}||^2$ where $\mathbf{A}$ is a $2N \times 12$ matrix depending on the 3D and 2D reference coordinates and $\mathbf{q}$ is the 12×1 vector $\mathbf{q} = [\mathbf{q}_1^\mathsf{T}, q_{14}, \mathbf{q}_2^\mathsf{T}, q_{24}, \mathbf{q}_3^\mathsf{T}, q_{34}]^\mathsf{T}$. The vector $\mathbf{q}$ is defined up to a scale factor, and the rank of $\mathbf{A}$ is equal 11 in general when the reference points are not co-planar. Also, the points should not form a specific *twisted cubic curve* in the space to avoid a singular case but this is highly unlikely for randomly chosen calibration points.

Least-square solution. The constraint $||\mathbf{q}_3||^2 = 1$ is invariant to changes in the WRF. Under this constraint, there exists the almost closed-form solution based on finding eigenvectors of a 3×3 matrix and inverting a 9×9 matrix. The constraint $(\mathbf{q}_1 \times \mathbf{q}_3) \bullet (\mathbf{q}_2 \times \mathbf{q}_3) = 0$ is also invariant to changes in the WRF. But this constraint does not allow for the closed-form solution.

The linear approach does not minimise the distance in the image between the 2D points $\mathbf{s}_i$ and the reprojected 3D points $\mathbf{S}_i$. The nonlinear minimisation criterion involves the total Cartesian distance explicitly:

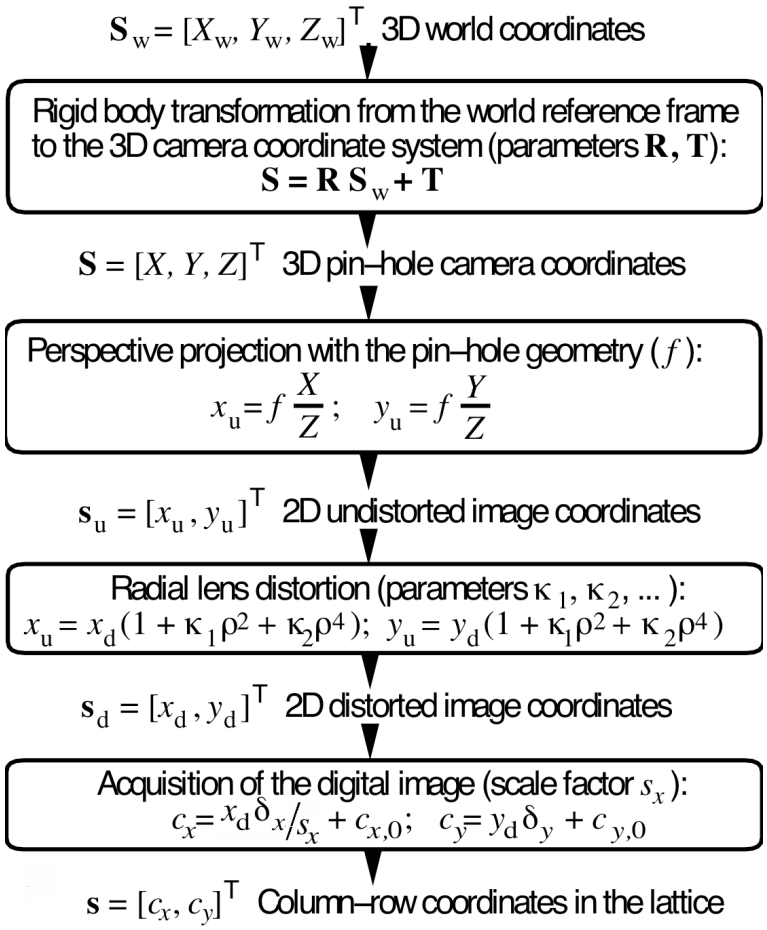
$$\sum_{i=1}^{N} \left|\left| \frac{\mathbf{q}_1^\mathsf{T}\mathbf{S}_i + q_{14}}{\mathbf{q}_3^\mathsf{T}\mathbf{S}_i + q_{34}} - c_{x,i} \right|\right|^2 + \left|\left| \frac{\mathbf{q}_2^\mathsf{T}\mathbf{S}_i + q_{24}}{\mathbf{q}_3^\mathsf{T}\mathbf{S}_i + q_{34}} - c_{y,i} \right|\right|^2$$

The total distance should be numerically minimised with respect to $\mathbf{q}$ subject to the constraint(s). Another way is to directly minimise it by the intrinsic and extrinsic parameters rather than the coefficients of the matrix $\mathbf{P}$.

The linear approach is very simple if it involves only the single constraint $||\mathbf{q}_3||^2 = 1$ and gives good results provided that the calibration pattern is chosen with a care. The non-linear method is much more robust with respect to noisy measurements of the reference coordinates. A need of a special calibration pattern in the field of view is cumbersome for many applications. There exist possibilities of estimating simultaneously both the camera parameters and the co-ordinates of the 3D points producing a given set of the corresponding image points $\mathbf{s}_i$; $i = 1, \ldots, N$.

2.4 Camera calibration: Tsai's scheme

This most popular calibration scheme is sketched step-by-step below.



Calibration steps 1 – 2. The Tsai's method requires at least 7 non-coplanar 3D reference points. The scale s_x is initially set to 1 and will be determined at step 4. First, co-ordinates $[x_{\rm d}, y_{\rm d}]$ in the distorted image are calculated from the lattice coordinates $[c_x, c_y]$ assuming that $[c_{x,0}, c_{y,0}]$ is the image center: $x_{\rm d} = s_x(c_x - c_{x,0})/\delta_x$ and $y_{\rm d} = (c_y - c_{y,0})/\delta_y$. Then 7 parameters transforming image coordinates into world co-ordinates are determined by replacing the denominator in $x_{\rm d} = s_x f \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_x}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_z}$ with its expression $r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_z = f\frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_y}{y_{\rm d}}$ derived from $y_{\rm d} = f\frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_y}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_z}$ and converting the resulting equation into the linear equation $x_{\rm d} = \mathbf{m}^{\mathsf{T}}\mathbf{L}$ assuming that $t_y \neq 0$. Here, $\mathbf{m} = [y_d X_w, y_d Y_w, y_d Z_w, y_d, -x_d X_w, -x_d Y_w, -x_d Z_w]^{\mathsf{T}}$ and $\mathbf{L} = \frac{1}{t_y}[s_x r_{11}, s_x r_{12}, s_x r_{13}, s_x t_x, r_{21}, r_{22}, r_{23}]^{\mathsf{T}}$.

More than 7 reference points lead to an over-determined system of equations $x_{d,i} = \mathbf{m}_i^{\mathsf{T}}\mathbf{L}; \quad i = 1, \ldots, N$, or $\mathbf{X} = \mathbf{M}\mathbf{L}$ where $\mathbf{X}$ is the $N \times 1$ vector of $x_{d,i}$ values and $\mathbf{M}$ is the $N \times 7$ matrix with the row vectors $\mathbf{m}_i^{\mathsf{T}}$. This system can be solved by using the pseudo-inverse technique called **Moore-Penrose inverse**[1]: $\mathbf{L} = \left(\mathbf{M}^{\mathsf{T}}\mathbf{M}\right)^{-1}\mathbf{M}^{\mathsf{T}}\mathbf{X}$.

Calibration steps 3-4. The Y-coordinate of the translation vector is computed using the parameter vector $\mathbf{L}$ obtained at step 2 and the orthonormality property of the rotation matrix $\mathbf{R}$: $r_{21}^2 + r_{22}^2 + r_{23}^2 = 1 \rightarrow |t_y| = \frac{1}{\sqrt{a_5^2 + a_6^2 + a_7^2}}$ where a_i denotes the i-th component of the above parameter vector $\mathbf{L} = [a_1, a_2, \ldots, a_7]^{\mathsf{T}}$. Then the scaling factor s_x is determined using the same orthonormality property of $\mathbf{R}$: $s_x = |t_y|\sqrt{a_1^2 + a_2^2 + a_3^2}$. The sign of t_y is obtained by choosing the reference 3D point whose image position is the most distant from the principal point (image center) and computing the parameters $r_{11} = a_1 t_y$, $r_{12} = a_2 t_y$, $r_{13} = a_3 t_y$, $r_{21} = a_5 t_y$, $r_{22} = a_6 t_y$, $r_{23} = a_7 t_y$, and $t_x = a_4 t_y$. The signs $\mathrm{sign}\{r_{11}X_w + r_{12}Y_w + r_{13}Y_w + t_x\}$ and $\mathrm{sign}\{r_{21}X_w + r_{22}Y_w + r_{23}Y_w + t_y\}$ of the computed coordinates of the projected point are compared to the signs of the actual image coordinates x, y. If the signs do not coincide, the sign of t_y should be inverted.

Calibration step 5 recalculates the 6 components of the rotation matrix $\mathbf{R}$ and the X-component of the translation vector $\mathbf{t}$: $r_{11} = a_1\frac{t_y}{s_x}$, $r_{12} = a_2\frac{t_y}{s_x}$, $r_{13} = a_3\frac{t_y}{s_x}$, $r_{21} = a_5 t_y$, $r_{22} = a_6 t_y$, $r_{23} = a_7 t_y$, and $t_x = a_4\frac{t_y}{s_x}$. The remaining 3 components of the rotation matrix are calculated using the inner vector (or cross) product of its first two rows: $r_{31} = \lambda\,(r_{12}r_{23} - r_{22}r_{13})$, $r_{32} = \lambda\,(r_{13}r_{21} - r_{23}r_{11})$, and $r_{31} = \lambda\,(r_{11}r_{22} - r_{21}r_{12})$ where the factor λ is given by the orthonormality property $r_{31}^2 + r_{32}^2 + r_{33}^2 = 1$.

[1]The Moore-Penrose inverse is obtained as follows. Let $\mathbf{A}\mathbf{x} = \mathbf{b}$ be an over-determined system of linear equations where $\mathbf{A}$ is a matrix $n \times m$, $\mathbf{x}$ is a vector $m \times 1$, and $\mathbf{b}$ is a vector $n \times 1$. The pseudo-inverse matrix for this system is obtained by solving the least-square problem $\min_{\mathbf{x}} D(\mathbf{x})$ where $D(\mathbf{x}) = ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2 \equiv \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{b} + \mathbf{b}^{\mathsf{T}}\mathbf{b}$.. The minimisation yields $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}} = 0$, or $(\mathbf{A}^{\mathsf{T}}\mathbf{A})\mathbf{x} - \mathbf{A}^{\mathsf{T}}\mathbf{b} = 0$. Thus the solution is: $\mathbf{x} = (\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{b}$ if the square $m \times m$ matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ is of rank m.

Calibration step 6 approximates the focal length f and the Z-coordinate of the translation vector accounting for no lens distortion. Projection relation for each reference point i: $y_{\mathrm{d},i} = f\frac{r_{21}X_{w,i}+r_{22}Y_{w,i}+r_{23}Z_{w,i}+t_y}{r_{31}X_{w,i}+r_{32}Y_{w,i}+r_{33}Z_{w,i}+t_z}$ is rewritten as the linear equation with respect to f and t_z: $[U_{y,i},\ -y_{\mathrm{d},i}][f,t_z]^{\mathsf{T}} = U_{z,i}y_{\mathrm{d},i}$ where $U_{y,i} = r_{21}X_{w,i} + r_{22}Y_{w,i} + r_{23}Z_{w,i} + t_y$ and $U_{z,i} = r_{31}X_{w,i} + r_{32}Y_{w,i} + r_{33}Z_{w,i}$.

More than 2 reference points i result in the over-determined system of equations: $\mathbf{M}[f,t_z]^{\mathsf{T}} = \mathbf{m}$ where

$$\mathbf{M}^{\mathsf{T}} = \begin{bmatrix} U_{y,1} & U_{y,2} & \dots & U_{y,n} \\ -y_{\mathrm{d},1} & -y_{\mathrm{d},2} & \dots & -y_{\mathrm{d},n} \end{bmatrix}$$

and $\mathbf{m}^{\mathsf{T}} = \begin{bmatrix} U_{z,1}y_{\mathrm{d},1} & U_{z,2}y_{\mathrm{d},2} & \dots & U_{z,n}y_{\mathrm{d},n} \end{bmatrix}$. This system has to be solved by the pseudo-inverse technique: $[f,t_z]^{\mathsf{T}} = (\mathbf{M}^{\mathsf{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathsf{T}}\mathbf{m}$. More accurate solutions can be obtained with the steepest descent optimisation that starts from the already determined approximate values.

Practical calibration procedure. The 3×4 theoretical calibration matrix that combines a rigid body (Euclidean motion) transformation $(\mathbf{R},\mathbf{T})$ from the WRF (world reference frame) to the CRF (camera reference frame) with projection onto the image plane (f) and integrates sensor specifications $(d'_{\mathrm{x}}, d_{\mathrm{y}})$ is determined as follows: (*i*) metric measurements for a set of real 3D points and corresponding pixel locations in the image plane are obtained using a specially designed calibration object (e.g., the cube in Fig. 2.8) with known co-ordinates of calibration points in the WRF, (*ii*) the calibration matrix coefficients are computed using the point-to-point correspondence between these measurements (as an optional step, the intrinsic and extrinsic camera parameters may be computed from these coefficients), and (*iii*) calibration errors are estimated by using a radius of ambiguity in ray tracing, that is, a distance between the back projected image point on the test plane and the initial measured 3D point in the same plane, or an accuracy of the 3D coordinate measurements obtained through stereo triangulation using the calibrated cameras. The calibration object in Fig. 2.8 should have both coplanar and non-coplanar points with precisely determined geometry and 3D coordinates. After feature points (patches, corners or predefined patterns) are manually or automatically extracted from the image, the Tsai's calibration can be performed.

A simpler calibration scheme estimates directly the 3×4 projection matrix that transforms a 3D point $(x,y,z)^{\mathsf{T}}$ in the WRF into its pixel position $(u,v)^{\mathsf{T}}$ in the digital image:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

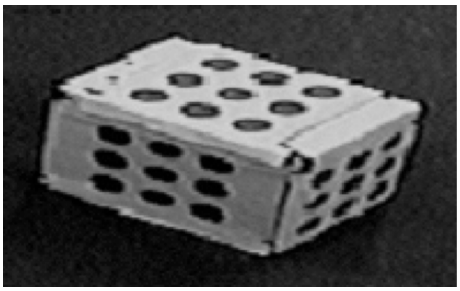


Figure 2.8: Calibration object.

where $s \neq 0$ and the matrix components are denoted for brevity $a, \ldots, k$. These latter relate to both intrinsic and extrinsic parameters of the camera. A linearised approximate solution is obtained by substituting $s = ix + jy + kz + 1$ into the two other equalities $su = ax + by + cz + d$ and $sv = ex + fy + gz + h$ so that

$$\begin{cases} ax + by + cz + d - ixu - jyu - kzu &= u \\ ex + fy + gz + h - ixv - jyv - kzv &= v \end{cases}$$

For a given set of N points with the known (or measured) $3D$ coordinates $(x_i, y_i, z_i)^{\mathsf{T}}$ and corresponding pixel positions (u_i, v_i), the following overdetermined linear system for the unknown components $a, \ldots, k$ holds:

$$\begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -x_1u_1 - y_1u_1 - z_1u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -x_1v_1 - y_1v_1 - z_1v_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -x_2u_2 - y_2u_2 - z_2u_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -x_2v_2 - y_2v_2 - z_2v_2 \\ x_3 & y_3 & z_3 & 1 & 0 & 0 & 0 & 0 & -x_3u_3 - y_3u_3 - z_3u_3 \\ 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 & 1 & -x_3v_3 - y_3v_3 - z_3v_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -x_Nu_N - y_Nu_N - z_Nu_N \\ 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -x_Nv_N - y_Nv_N - z_Nv_N \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ \vdots \\ \vdots \\ u_N \\ v_N \end{pmatrix}$$

This system can be solved by the pseudo inverse techniques.

2.5 Binocular viewing

Let the coordinates of the points of a spatial scene be measured in a fixed 3D Cartesian WRF $OXYZ$. Let the 2D coordinates of the image pixels be measured in the fixed

2D Cartesian systems $o_1x_1y_1z_1$ and $o_2x_2y_2z_2$, respectively. Figure 2.9 shows geometry of binocular viewing of a 3D scene by two cameras with arbitrary positions and orientations. Here, the following notations are used: $\mathbf{O}_1 = (X_{o,1}, Y_{o,1}, Z_{o,1})$ and $\mathbf{O}_2 = (X_{o,2}, Y_{o,2}, Z_{o,2})$ denote optical centres of the first and second camera, $\mathbf{o}_1 = (x_{1,o}, y_{1,o})$ and $\mathbf{o}_2 = (x_{2,o}, y_{2,o})$ are principal points of the first and second images, that is, the traces of the optical axes of the first and the second cameras in the corresponding images, $\mathbf{e}_1$ is an epipolar point in the plane of the first image, that is, the projection of the second optical centre $\mathbf{O}_2$ onto this plane, and $\mathbf{e}_2$] is an epipolar point in the plane of the second image, that is, the projection of the first optical centre $\mathbf{O}_1$ onto this plane. Also, let $\mathbf{S} = (X, Y, Z)$ be an arbitrary 3D point. Then $\mathbf{s}_1$ denotes the projection of the point $\mathbf{S}$ onto the first image, and $\mathbf{s2}$ denotes the projection of the point $\mathbf{S}$ into the second image.

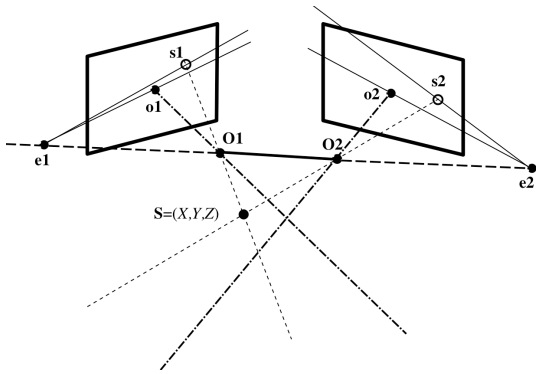


Figure 2.9: Binocular viewing and fundamental matrix.

The line segment $\overline{\mathbf{O}_1\mathbf{O}_2}$ between the optical centres is called the **stereo baseline**. The binocular viewing with the two cameras is conveniently described in terms of the corresponding epipolar lines in the stereo pair of images. Let $\mathbf{s}$ denote the projection of a 3D point $\mathbf{S}$ onto an image plane. The epipolar line through the pixel $\mathbf{s}$ in this image plane is defined as a trace of the intersecting plane that contains the 3D point $\mathbf{S}$ and the baseline (that is, both the optical centres $\mathbf{O1}$ and $\mathbf{O2}$). Any spatial point in this plane is projected into the corresponding pair of the epipolar lines in the images, for instance, into the lines $\overline{\mathbf{e}_1\mathbf{s}_1}$ and $\overline{\mathbf{e}_2\mathbf{s}_2}$. Thus, an epipolar profile of the scene (that is, the profile of the scene in the intersecting plane $\mathbf{SO}_1\mathbf{O}_2$) is depicted by the corresponding epipolar lines in the images.

The epipolar geometry is outlined in Figure 2.10 where $\mathbf{s}_1, \mathbf{s}_2$ are the projections of a 3D point $\mathbf{S}$, and $\mathbf{e}_1, \mathbf{e}_2$ are **epipoles**, or the projections of the centre $\mathbf{O}_1$ and $\mathbf{O}_2$ onto the second and first image, respectively. The lines $\overline{\mathbf{e}_1\mathbf{s}_1}$ and $\overline{\mathbf{e}_2\mathbf{s}_2}$ are called the corresponding **epipolar lines**.

For binocular stereo viewing, the following symmetric epipolar constraint holds. For a given point $\mathbf{s}_1$ ($\mathbf{s}_2$) in the plane of the stereo image 1 (2), all the possible stereo matches in the plane of another image 2 (1) lie on the epipolar line through the epipole $\mathbf{e2}$ ($\mathbf{e1}$),

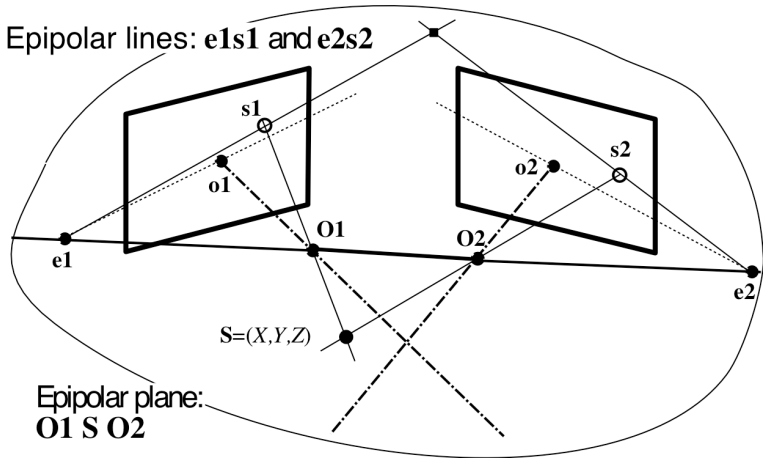
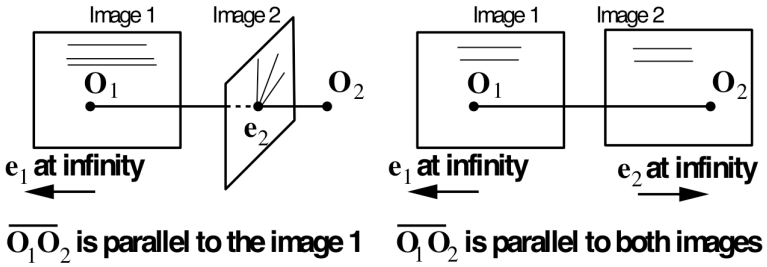


Figure 2.10: Epipolar geometry.

respectively. The corresponding epipolar lines are the intersections of the plane $\mathbf{SO_1O_2}$ with the image planes. The parallel epipolar lines are in a special case of a so-called horizontal stereo pair:



Epipolar relations between the image points are as follows. Given the two cameras with the projection matrices $\mathbf{P}_i = [\mathbf{Q}_i\ \mathbf{q}_i]$; $i = 1, 2$, a 3D point $\mathbf{S}$ relates to the corresponding image points as $\mathbf{s}_1 = \mathbf{P}_1\mathbf{S}$ and $\mathbf{s}_2 = \mathbf{P}_2\mathbf{S}$.

As shown earlier, the optical centre $\mathbf{O}_i = -\mathbf{Q}_i\mathbf{q}_i$. Thus the epipole $\mathbf{e}_j$; $j \neq i$, given by the relationship

$$\mathbf{e}_j = \mathbf{P}_j \begin{bmatrix} -\mathbf{Q}_i^{-1}\mathbf{q}_i \\ 1 \end{bmatrix},$$

is one of the points of each epipolar line.

Another point $\mathbf{D}_i$ can be chosen at infinity of the optical ray: $\overline{\mathbf{O}_i\mathbf{s}_i}$, that is, $\mathbf{D}_i = \mathbf{Q}_i^{-1}\mathbf{s}_i$. The image $\mathbf{d}_j$ of this point in the second image plane is given by

$$\mathbf{d}_j = \mathbf{Q}_j\mathbf{Q}_i^{-1}\mathbf{s}_i.$$

Fundamental matrix. Given the two points $\mathbf{e}_2$ and $\mathbf{d}_2$, the epipolar line in the image plane 2 in the homogeneous coordinates is represented by the vector cross product $\mathbf{e}_2 \times \mathbf{d}_2$.

The cross product $\mathbf{e}_2 \times \mathbf{d}_2$ can be written as $\mathbf{F}\mathbf{s}_1$ where $\mathbf{F}$ is a 3×3 *fundamental matrix*. As shown by Faugeras [4] (Chapter 6, p. 172) any pixel $\mathbf{s}_2$ on the epipolar line of $\mathbf{s}_1$ satisfies the equation:

$$\mathbf{s}_2^{\mathrm{T}}\mathbf{F}\mathbf{s}_1 = 0.$$

where T, as usually, indicates the transposition. This equation is called the **Longuet-Higgins equation**. It suggests that the parameters of the epipolar line of $\mathbf{s}_1$ are given by the vector $\mathbf{s}_2^{\mathrm{T}}\mathbf{F}$ as well as the parameters of the epipolar line of $\mathbf{s}_2$ are given by the vector $\mathbf{F}\mathbf{s}_1$.

Let homogeneous coordinate vectors

$$\left(\mathbf{s}_{k,j} = \begin{bmatrix} x_{k,j} \\ y_{k,j} \\ 1 \end{bmatrix} : \; j = 1, 2 \right)$$

denote the k-th pair of corresponding points in a stereo pair of images (the indices $j = 1$ and $j = 2$ represent the left and the right image of the stereo pair, respectively).

The fundamental matrix relationship between the homogeneous coordinates of the corresponding points:

$$\mathbf{s}_{k,2}^T\mathbf{F}\mathbf{s}_{k,1} = 0. \tag{2.1}$$

means that any point $\mathbf{s}_{k,2}$ of the right image specifies in the left image an epipolar line which the corresponding point $\mathbf{s}_{k,1}$ lies on. The line has the parameterss $\mathbf{s}_{k,2}^T\mathbf{F}$. Alternatively, the point $\mathbf{s}_{k,1}$ specifies in the right image the parameters $\mathbf{F}\mathbf{s}_{k,1}$ of the corresponding epipolar line which the point $\mathbf{s}_{k,2}$ lies on.

The parameters of the epipolar lines can be represented by the coordinates of the epipoles $\mathbf{e}_1 = [x_{e,1}, y_{e,1}]^T$ and $\mathbf{e}_2 = [x_{e,2}, y_{e,2}]^T$ in the images. In this case, Eq. (2.1) takes the following form:

$$\begin{array}{rcl} a_1(x_{k,1} - x_{e,1})(x_{k,2} - x_{e,2}) + a_2(y_{k,1} - y_{e,1})(x_{k,2} - x_{e,2}) & + & \\ a_4(x_{k,1} - x_{e,1})(y_{k,2} - y_{e,2}) + a_5(y_{k,1} - y_{e,1})(y_{k,2} - y_{e,2}) & = & 0, \end{array} \tag{2.2}$$

Thus, the matrix $\mathbf{F}$ in Eq. (2.1) depends on four parameters $\mathbf{a} = [a_1, a_2, a_4, a_5]^T$ and four coordinates $\mathbf{e} = [x_{e,1}, y_{e,1}, x_{e,2}, y_{e,2}]^T$ of the epipoles as follows:

$$\mathbf{F} = \left(\begin{array}{c|c|c} a_1 & a_2 & -x_{e,1} \cdot a_1 - y_{e,1} \cdot a_2 \\ \hline a_4 & a_5 & -x_{e,1} \cdot a_4 - y_{e,1} \cdot a_5 \\ \hline -x_{e,2} \cdot a_1 & -x_{e,2} \cdot a_2 & x_{e,1} \cdot x_{e,2} \cdot a_1 + y_{e,1} \cdot x_{e,2} \cdot a_2 \\ -y_{e,2} \cdot a_4 & -y_{e,2} \cdot a_5 & +x_{e,1} \cdot y_{e,2} \cdot a_4 + y_{e,1} \cdot x_{y,2} \cdot a_5 \end{array} \right). \tag{2.3}$$

It is easily seen that the matrix $\mathbf{F}$ in Eq. (2.3) has the rank 2.

Figures 2.11 – 2.13 show changes of the corresponding epipolar lines in the images if the parameters $\mathbf{a}$ of the fundamental matrix are changed but the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$ are fixed (compare Figs. 2.11 and 2.12) and if the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$ are moved but the parameters $\mathbf{a}$ are fixed (compare Figs. 2.11 and 2.13). Thus it is in principle possible to sequentially change the fundamental matrix $\mathbf{F}$ under the fixed epipoles and change each epipole under the fixed matrix and the other epipole for minimising the total squared distance between a given set of the corresponding pixels and the epipolar lines produced by these pixels.

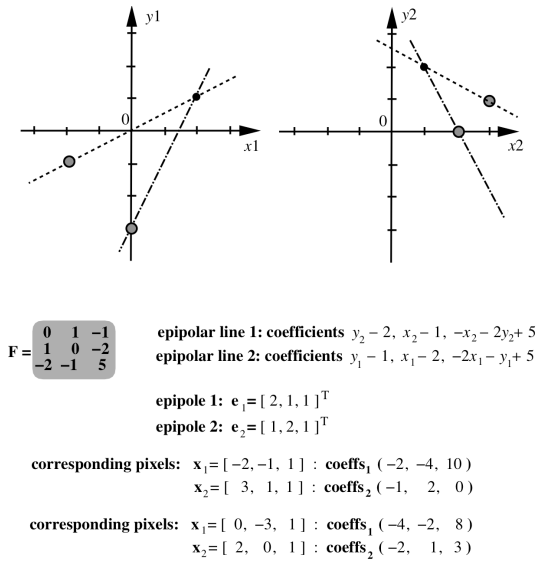


Figure 2.11: Corresponding pixels $\mathbf{s}_{j,k}$, $j = 1, 2$; $k = 1, 2$, and epipolar lines under given parameters $\mathbf{a}, \mathbf{e}$.

To represent the squared distance between a pixel and an epipolar line generated by the corresponding pixel, the squared left side of Eq. (2.1) can be rewritten in an equivalent form:

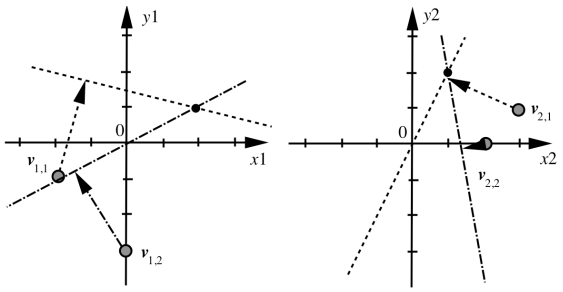
$$\left(\mathbf{s}_{k,2}^T \mathbf{F} \mathbf{s}_{k,1}\right)^2 \equiv \mathbf{a}^T \mathbf{\Phi}_k(\mathbf{e}) \mathbf{a} \tag{2.4}$$

with the following 4×4 matrix $\mathbf{\Phi}_k(\mathbf{e})$:

$$\mathbf{\Phi}_k(\mathbf{e}) = \mathbf{f}_k(\mathbf{e}) \cdot \mathbf{f}_k^T(\mathbf{e}) \tag{2.5}$$

where the vector $\mathbf{f}_k(\mathbf{e})$ is as follows:

$$\mathbf{f}_k(\mathbf{e}) = \begin{bmatrix} (x_{k,1} - x_{e,1}) \cdot (x_{k,1} - x_{e,1}) \\ (y_{k,1} - y_{e,1}) \cdot (x_{k,2} - x_{e,2}) \\ (x_{k,1} - x_{e,1}) \cdot (y_{k,2} - y_{e,2}) \\ (y_{k,1} - y_{e,1}) \cdot (y_{k,2} - y_{e,2}) \end{bmatrix}. \tag{2.6}$$



$$\mathbf{F} = \begin{bmatrix} 1 & 2 & -4 \\ 1 & 0 & -2 \\ -3 & -2 & 8 \end{bmatrix}$$

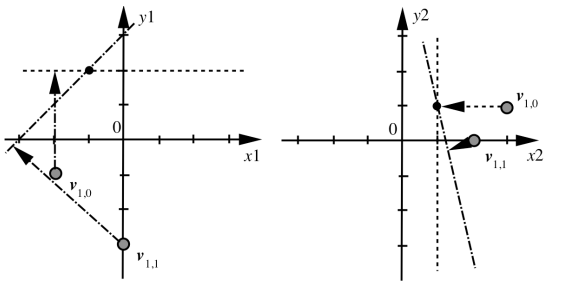
ep. line 1: coeffs $x_2 + y_2 - 3,\ 2x_2 - 2,\ -4x_2 - 2y_2 + 8$
ep. line 2: coeffs $x_1 + 2y_1 - 4,\ x_1 - 2,\ -3x_1 - 2y_1 + 8$

epipole 1: $\mathbf{e}_1 = [\,2, 1, 1\,]^T$
epipole 2: $\mathbf{e}_2 = [\,1, 2, 1\,]^T$

corresponding pixels: $\mathbf{x}_1 = [\,-2, -1, 1\,]$: **coeffs** $_2$ $(\,-8, -4, 16\,)$
$\mathbf{x}_2 = [\,3, 1, 1\,]$: **coeffs** $_1$ $(\,1, 4, -6\,)$

corresponding pixels: $\mathbf{x}_1 = [\,0, -3, 1\,]$: **coeffs** $_2$ $(-10, -2, 14\,)$
$\mathbf{x}_2 = [\,2, 0, 1\,]$: **coeffs** $_1$ $(\,-1, 2, 0\,)$

Figure 2.12: Changes of the epipolar lines in Fig. 2.11 for the new parameters a.



$$\mathbf{F} = \begin{bmatrix} 0 & 1 & -2 \\ 1 & 0 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

epipolar line 1: coefficients $y_2 - 1,\ x_2 - 1,\ -2x_2 + y_2 + 1$
epipolar line 2: coefficients $y_1 - 2,\ x_1 + 1,\ -x_1 - y_1 + 1$

epipole 1: $\mathbf{e}_1 = [\,-1, 2, 1\,]^T$
epipole 2: $\mathbf{e}_2 = [\,1, 1, 1\,]^T$

corresponding pixels: $\mathbf{x}_1 = [\,-2, -1, 1\,]$: **coeffs** $_1$ $(\,0, 2, -4\,)$
$\mathbf{x}_2 = [\,3, 1, 1\,]$: **coeffs** $_2$ $(\,-3, 0, 3\,)$

corresponding pixels: $\mathbf{x}_1 = [\,0, -3, 1\,]$: **coeffs** $_1$ $(\,-1, 1, -3\,)$
$\mathbf{x}_2 = [\,2, 0, 1\,]$: **coeffs** $_2$ $(\,-5, 1, 4\,)$

Figure 2.13: Changes of the epipolar lines in Fig. 2.11 for the new epipoles' (e) position..

Thus the squared distance $d_{k,1}(\mathbf{a},\mathbf{e})$ of a pixel $\mathbf{s}_{k,1}$ from the epipolar line that corresponds to the pixel $\mathbf{s}_{k,2}$ and the like distance $d_{k,2}(\mathbf{a},\mathbf{e})$ of a pixel $\mathbf{s}_{k,2}$ from the epipolar line that corresponds to the pixel $\mathbf{s}_{k,1}$ can be represented as follows:

$$\begin{array}{l} d_{k,1}(\mathbf{a},\mathbf{e}) = \frac{\mathbf{a}^T\mathbf{\Phi}_k(\mathbf{e})\mathbf{a}}{\mathbf{a}^T\mathbf{C}_{k,1}(\mathbf{e}_2)\mathbf{a}}; \\ d_{k,2}(\mathbf{a},\mathbf{e}) = \frac{\mathbf{a}^T\mathbf{\Phi}_k(\mathbf{e})\mathbf{a}}{\mathbf{a}^T\mathbf{C}_{k,2}(\mathbf{e}_1)\mathbf{a}}, \end{array} \tag{2.7}$$

where the denominators present the normalizing factors:

$$\mathbf{a}^T\mathbf{C}_{k,1}(\mathbf{e}_2)\mathbf{a} \equiv \\ (a_1\cdot(x_{k,2}-x_{e,2})+a_2\cdot(y_{k,2}-y_{e,2}))^2+(a_4\cdot(x_{k,2}-x_{e,2})+a_5\cdot(y_{k,2}-y_{e,2}))^2$$

and

$$\mathbf{a}^T\mathbf{C}_{k,2}(\mathbf{e}_1)\mathbf{a} \equiv \\ (a_1\cdot(x_{k,1}-x_{e,1})+a_4\cdot(y_{k,1}-y_{e,1}))^2+(a_2\cdot(x_{k,1}-x_{e,1})+a_5\cdot(y_{k,1}-y_{e,1}))^2 .$$

Here, the matrices $\mathbf{C}_{k,1}(\mathbf{e}_2)$ and $\mathbf{C}_{k,2}(\mathbf{e}_1)$ have the following obvious form:

$$\mathbf{C}_{k,1}(\mathbf{e}_2) = \left(\begin{array}{c|c|c|c} (x_{k,2}-x_{e,2})^2 & (x_{k,2}-x_{e,2})\cdot(y_{k,2}-y_{e,2}) & 0 & 0 \\ \hline (x_{k,2}-x_{e,2})\cdot(y_{k,2}-y_{e,2}) & (y_{k,2}-y_{e,2})^2 & 0 & 0 \\ \hline 0 & 0 & (x_{k,2}-x_{e,2})^2 & (x_{k,2}-x_{e,2})\cdot(y_{k,2}-y_{e,2}) \\ \hline 0 & 0 & (x_{k,2}-x_{e,2})\cdot(y_{k,2}-y_{e,2}) & (y_{k,2}-y_{e,2})^2 \end{array}\right)$$

and

$$\mathbf{C}_{k,2}(\mathbf{e}_1) = \left(\begin{array}{c|c|c|c} (x_{k,1}-x_{e,1})^2 & 0 & (x_{k,1}-x_{e,1})\cdot(y_{k,1}-y_{e,1}) & 0 \\ \hline 0 & (x_{k,1}-x_{e,1})^2 & 0 & (x_{k,1}-x_{e,1})\cdot(y_{k,1}-y_{e,1}) \\ \hline (x_{k,1}-x_{e,1})\cdot(y_{k,1}-y_{e,1}) & 0 & (y_{k,1}-y_{e,1})^2 & 0 \\ \hline 0 & (x_{k,1}-x_{e,1})\cdot(y_{k,1}-y_{e,1}) & 0 & (y_{k,1}-y_{e,1})^2 \end{array}\right) .$$

Parameter normalization. Basic relations in Eqs. (2.1) and (2.7) suggest that components of the fundamental matrix should be normalized to exclude the singular case of $\mathbf{F}=\mathbf{0}$.

It should be noted that an ideal horizontal stereopair with the epipolar lines $y_1 = y_2 = const$ parallel to the x-axis of the images has the following fundamental matrix:

$$\mathbf{F} = \left(\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & 0 & \frac{1}{\sqrt{2}} \\ \hline 0 & -\frac{1}{\sqrt{2}} & 0 \end{array}\right) \tag{2.8}$$

with the parameters $\mathbf{a} = \mathbf{0}$ and the parameters $\mathbf{e} = [-\infty, c_1, -\infty, c_2]^T$ where the constants c_j may have arbitrary values.

Thus, it is impossible to normalize only the parameters $\mathbf{a}$, and the normalization has to take account of all the components of Eq. (2.3) which are present in the normalizing factors of the distances of Eq. (2.7). In other words, if $\mathbf{F} = (F_{i,j})_{i,j=1}^{3}$ then all the components, except for the component $F_{3,3}$, should be normalized.

Appendix B presents more detail on estimating the epipolar geometry based on the fundamental matrix.

Chapter 3

Colour Discrimination

A colour is a subjective human perception of visible light depending on an intensity and a set of wavelengths associated with the electromagnetic spectrum. This subjective visual characteristic describes how perceived electromagnetic radiation $F(\lambda)$ is distributed in the range of wavelengths λ of visible light in the range of wavelengths [380 nm ... 780 nm]. The composition of wavelengths specifies *chrominance* of visible light for human visual system. The chrominance has two attributes, *hue* and *saturation*. The hue is characterised by the dominant wavelength(s) in the composition, and the saturation measures the purity of a colour. A pure colour has 100% of saturation, whereas all shades of colourless (grey) light, e.g. white light, have 0% of saturation.

The sensed colour varies considerably with 3D surface orientation, camera viewpoint, and illumination of the scene, e.g., positions and spectra of illuminating sources. Also, human colour perception is quite subjective as regarding perceptual similarity. To design formal colour descriptors, one should specify a colour space, its partitioning, and how to measure similarity between colours. A colour space is a multidimensional space of colour components. Human colour perception combines the three primary colours: red (R) with the wavelength $\lambda = 700$nm, green (G) with the wavelength $\lambda = 546.1$nm, and blue (B) with the wavelength $\lambda = 435.8$nm. Nearly any visible wavelength λ is sensed as a colour obtained by a linear combination of the three primary colours (R, G, B) with the particular weights $c_R(\lambda)$, $c_G(\lambda)$, and $c_B(\lambda l)$: $F(\lambda) = Rc_R(\lambda) + Gc_G(\lambda) + Bc_B(\lambda)$.

3.1 Colour models

The simplest colour model is a weighted sum of the three primary colours (Fig. 3.1). An additive mixture based on the RGB (Red, Green, Blue) colours is used to display images. A subtractive mixture based on the CMYK (Cyan, Yellow, Magenta, blacK) colours allows for printing images.

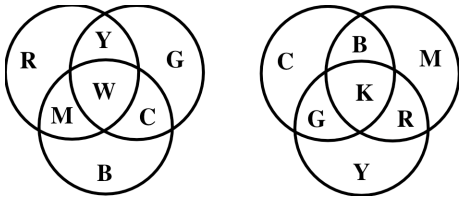


Figure 3.1: Colour models.

RGB colour model assumes by the international standard the following spectral characteristics of the primary colours: R 700 nm, G 546.1 nm, and B 435.8 nm. The RGB cube (see Fig. 3.2) has all grey values on the main diagonal line $(0,0,0)$ (black) — $(1,1,1)$ (white), and the primary colours are on the nodes R = $(1,0,0)$; G = $(0,1,0)$; B = $(0,0,1)$; C = $(0,1,1)$; M = $(1,0,1)$; Y = $(1,1,0)$; blacK = $(0,0,0)$; White = $(1,1,1)$ The RGB colour coordinates are strongly interdependent and describe not only inherent colour properties of an object, but also variations of illumination and other external factors.

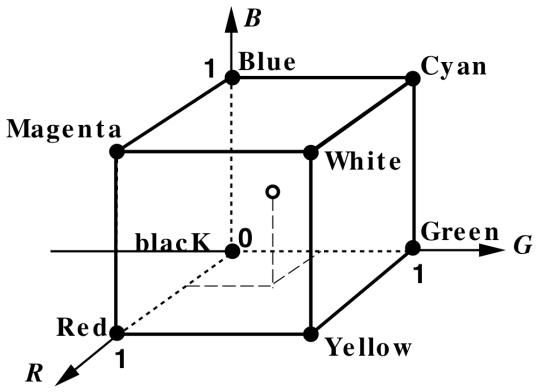


Figure 3.2: RGB cube.

More convenient colour representation is provided by **independent** (or **opponent**) colour axes, e.g. $(R+G+B)/3, 0.5R-0.5G, -0.5R-0.5G+B)$ where the first axis indicates intensity or luminance of the optical colour signal and two other axes describe its chrominance, or several other "luminance - chrominance" representations. Such a representation separates the luminance of the signal (e.g., $(R+B+G)/3$) from the two chrominance components in the co-ordinate plane orthogonal to the luminance axis making the chrominance components invariant to changes in illumination intensity and shadows. But although these linear colour transforms are computationally simple, the resulting colour spaces are neither uniform, nor natural.

HSI / HSV colour model. The HSI (hue - saturation - intensity) or, what is the same, HSV (hue - saturation - value) colour space is obtained by a non-linear transformation of the RGB space and provides more adequate representation of colours. The brightness value(or intensity) $I = (R+G+B)/3$ acts as the main axis orthogonal to the chrominance plane. The saturation S and hue H are the radius and angle, respectively, of the polar coordinates in the chrominance plane with the origin in the trace of the main axis (see Fig. 3.3).

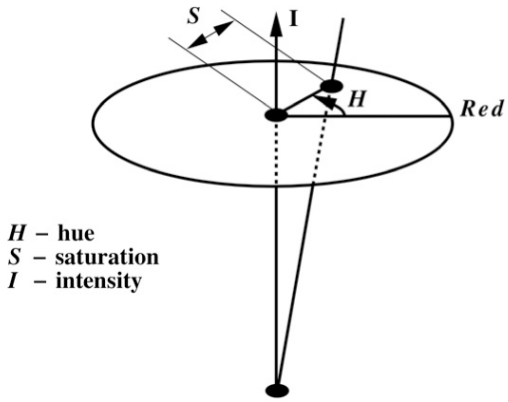


Figure 3.3: RGB to HSI transformation.

This representation is perceptually rather uniform and it is closely related to the way the human vision perceives colour images. Because of invariance to the object orientation with respect to illumination and camera viewing direction, the colour hue might be more suitable for colour object retrieval. But the conversion between the RGB and HSI colour coordinates is rather complicated:

$$\begin{aligned} H &= \begin{cases} \delta & \text{if} \quad B < G \\ 360 - \delta & \text{otherwise} \end{cases} \\ S &= 1 - 3\,\frac{\min\{R,G,B\}}{R+G+B} \\ I &= \frac{R+G+B}{3} \end{aligned} \tag{3.1}$$

where[1] $\delta = \cos^{-1}\left(\dfrac{0.5\,((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-G)(G-B)}}\right)$ in the interval $[0, 180^\circ]$.

[1] $\cos^{-1} z$ denotes the angles θ such that $\cos\theta = z$.

3.2 Vector quantisation of a colour space

Generally, the colour space is much more detailed than human vision requires for representing natural objects, and every image or video clip does not use simultaneously all the perceivable colours . With 256 signal levels for each RGB colour component, the RGB cube splits into $2^{24} = 16,277,216$ individual colours whereas most of scenes involve only hundreds and rarely thousands of different colours. Thus the discrete colour space can be considerably compressed by proper colour quantisation.

With respect to accuracy of representing colours of each individual image, a scalar quantisation of colour spaces, that is, a separate quantisation of each colour dimension, ranks below an adaptive vector quantisation. Generally, the **vector quantisation** maps a whole d-dimensional vector space into a finite set

$$\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$$

of representative d-dimensional vectors. The set $\mathbf{C}$ is usually called a **codebook**, and its elements are called **code words**. In colour quantisation, $d = 3$, and each code word $\mathbf{c}_k$ is a representative colour. The codebook $\mathbf{C}$ representing a collection of K colours is usually called a colour **gamut**, or a **palette**. The vector quantisation partitions the whole 3D colour space into K disjoint subsets, one per code word. All the colours belonging to the same subset are represented by, or quantised to the same code word $\mathbf{c}_k$. A perceptually good palette contains code words that closely approximate colours in the corresponding subsets so that each subset contains the visually similar colours.

Many digital graphics formats use one or another form of vector quantisation to compress the colour images. The palette for an image or an ensemble of images is usually built by statistical averaging and clustering of the colours at hand. Any conventional multidimensional clustering method, such as K-means, fuzzy K-means, or EM (Expectation - Maximisation) clustering discussed below in Chapter 4, can be used in principle for the colour quantisation.

One popular vector quantisation algorithm iteratively doubles the number of codewords until a prescribed number of them, say, 64, 128, or 256, is formed. Each iteration t creates $K_t = 2^t$ cluster centres (codewords)

$$\mathbf{C}_t = \{\mathbf{c}_{k,t} : \ k = 1, \ldots, K_t\}$$

When $t = 0$, the process starts with a single centre $\mathbf{c}_{1,0}$ that averages colour vectors over an image. At each next iteration, $t = 1, 2, \ldots$, every previous cluster centre $\mathbf{c}_{k,t-1}$; $k = 1, ..., K^{t-1}$, splits into the two new centres as follows:

1. each current code word $\mathbf{c}_{k,t-1}$ splits into the two new provisional code words, $\mathbf{c}_{\mathrm{pr}:k,t}$ and $\mathbf{c}_{\mathrm{pr}:K_{t-1}+k,t}$;

2. each colour vector in the image is assigned to the closest new cluster (the closeness between a colour vector and a code word is determined using a particular metrics in the colour space); and

3. the new code words (cluster centres) $\mathbf{c}_{k,t}$ and $\mathbf{c}_{K_{t-1}+k,t}$ are formed by averaging the colour vectors assigned to each such cluster.

Strategies of how to split one code word differ in different implementations of this simple clustering algorithm, e.g. a multiplication to the two constant factors: $(1+w)\mathbf{c}$ and $(1-w)\mathbf{c}$ where w; $0 < w < 1$, is a fixed constant, or a shift of each current centre to and from the most distant signal $\mathbf{g}$ in the cluster:

$$\mathbf{c} + w \cdot (\mathbf{g} - \mathbf{c}) \quad \text{and} \quad \mathbf{c} - w \cdot (\mathbf{g} - \mathbf{c})$$

or so forth.

3.3 Colour descriptors

Colour descriptors of images can be global and local. The former ones specify the overall colour content of the image but with no information about the spatial distribution of colours. Local descriptors relate to particular image regions. Most popular descriptors are colour histograms.

Colour histograms A colour histogram describes the distribution of colours within a whole image or a specified region. As a pixel-wise characteristic, the histogram is invariant to rotation, translation, and scaling of an object. At the same time, the histogram does not capture spatial relationships among colours. A quantised HSI (or HSV) colour space is typically used to represent the colour in order to partially make the histograms invariant to illumination and object viewpoints. In the HSI colour space, an Euclidean or similar component-wise distance between the components specifies colour similarity quite well.

A normalised colour histogram $\mathbf{h}(\text{image}) = (h_k(\text{image}) : \ k = 1, \ldots, K)$ is a K-dimensional vector such that each component $h_k(\text{image})$ represents the relative number of pixels of colour $\mathbf{c}_k$ in the image, that is, the fraction of pixels that are most similar to the corresponding representative colour:

$$h_k(\text{image}) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \delta(\text{image}(x,y), \mathbf{c}_k)$$

where the Kroneker-type δ-function is equal 0 or 1 if the colour $\text{image}(x,y)$ in an image pixel (x,y) does not coincide or coincides with the code word $\mathbf{c}_k$, respectively. The components of the normalised histogram are the relative frequencies of the individual colours (code words):

$$\sum_{k=1}^{K} h_k(\text{image}) = 1$$

To built such a colour histogram, the image colours should be transformed to an appropriate colour space and quantised according to a particular codebook of the size K. These colour histograms can be easily collected for individual image regions.

To detect regions that are similar in the overall colours, their colour histograms should be compared. The (dis)similarity of the two colour histograms, $\mathbf{h}$ and $\mathbf{h}'$, is measured by computing a distance between the histograms in the colour space. The chosen metric effects both effectiveness and computational complexity of retrieval. The effectiveness indicates to which extent the quantitative similarity match the perceptual, subjective one.

In the simplest case, the distance is based on the Minkowski metrics, such as the city-block or Euclidean distance between the relative frequencies of the corresponding colours, or on the *histogram intersection* proposed by Swain and Ballard:

$$\begin{aligned} D_{\text{city}-\text{block}}(\mathbf{h}, \mathbf{h}') &= \sum_{k=1}^{K} |h_k - h'_k| \\ D_{\text{Euclidean}}(\mathbf{h}, \mathbf{h}') &= \sum_{k=1}^{K} (h_k - h'_k)^2 \\ D_{\text{intersection}}(\mathbf{h}.\mathbf{h}') &= 1 - \sum_{k=1}^{K} \min\{h_k, h'_k\} \equiv \tfrac{1}{2} D_{\text{city}-\text{block}}(\mathbf{h}, \mathbf{h}') \end{aligned}$$

The above metrics comparing only the corresponding colour components between the histograms take no account of cross-relations of the different colour clusters. Thus the images with similar but not identical representative colours can be considered as dissimilar on the basis of the distance between the colour histograms. Quadratic-form metrics avoid this drawback by pairwise comparisons of all the component pairs:

$$D(\mathbf{h}, \mathbf{h}') = (\mathbf{h} - \mathbf{h}')^{\mathsf{T}} \, \mathbf{A} \, (\mathbf{h} - \mathbf{h}')$$

where $\mathbf{A} = [a_{ij}]$ is the positive definite symmetric matrix $K \times K$ with components $a_{ij} = a_{ji}$ specifying the dissimilarity between the code words $\mathbf{c}_i$ and $\mathbf{c}_j$ for the histogram components with indices i and j. To decrease the computational complexity of the quadratic-form metrics, only most significant components may be taken into account.

A special case of the quadratic-form metric is the Mahalanobis distance in which the dissimilarity matrix $\mathbf{A}$ is obtained by inverting the covariance matrix for a training set of colour histograms. Alternatively, the Mahalanobis distance can account for the covariance matrix of colours in a set of training images (then the colours that are dominant across all images and do not discriminate among different images will not effect the distance, as it should be). In the special case of uncorrelated histogram components when the covariance matrix is diagonal, the Mahalanobis distance reduces to a weighted Euclidean one. The weight of each squared difference of the histograms' components is inversely proportional to the variance of these components treated as random variables.

Other colour descriptors The colour information can be also represented using colour moments and colour sets. **Colour moments** are used sometimes as feature vectors in order to overcome quantisation effects of the colour histogram. Any colour distribution can

be characterised by its moments, and typically the low-order moments are most informative. Usually only the first few central moments, namely, the mean colour M_1, variance M_2, and skewness M_3, act as scalar features of the colour components:

$$\begin{aligned} M_{1,q} &= \sum_{k=1}^{K} h_k c_{q,k} \\ M_{S,q} &= \left(\sum_{k=1}^{K} h_k \left(c_{q,k} - M_{q,1} \right)^S \right)^{\frac{1}{S}} \end{aligned}$$

¡p¿ Here, q denotes the colour component (e.g., R, G, B or H, S, V) and $S = 2, 3, \ldots$, is the order of the moment. The similarity between the moments is measured usually by the Euclidean distance.

A **colour set** represents another reduced collection of colour features. The set is obtained by thresholding the colour histogram. All image colours are first quantised into a fixed relatively small number of colours in the HSI colour space, and then the colour set is defined as a subset of most characteristic colours. Two images with the same colour set are regarded as similar even if they have different relative amount of colours.

In particular, the colour HSI space can be partitioned into 166 characteristic colours as follows. The HSI space is considered as a cylinder with the axis representing the intensity that ranges from pure black (0) to pure white (1). The distance (radius) to the axis gives the saturation, or relative amount of presense of a colour, and the angle around the axis is the hue giving the chroma (tint, ot tone). The hue is represented with the finest resolution by a circular quantisation of the hue circle into 18 sectors (6 per each primary colour). Other colour components are represented with the coarser resolution by quantising each into three levels. In addition, the colourless greyscale signals are quantised into four levels. This gives in total 18 (H) $\times$ 3 (S) $\times$ 3 (I) + 4 (grey levels) = 166 disctinct colours.

But it should be noted that the colour histograms, moments, and sets do not describe spatial relationships among the neighbouring pixels.

3.4 Colour predicate for image segmentation

A colour predicate (CP) for skin detection and segmentation in an image was first proposed by Kjeldsen and Kender in [10]. In general, their procedure is presented below but with a number of changes to make it more convenient for implementation. In particular, CPs in [10, 15] have required from the user to identify skin regions in training images through manually drawn binary masks. A semi-automated method below for the training of the predicate uses a simplified logarithmic hue to threshold the training images. The logarithmic hue proposed by Lievin and Luthon [11] is not only simpler than the conventional hue but also is more robust to varying illumination. Rather than the initial 3-dimensional CP indexed with hue, saturation, and intensity values in each pixel, a reduced predicate with only the logarithmic Hue and Saturation indexes is used below. This

significantly reduces time for segmentation, while ensures still the sufficiently accurate skin detection.

Automatic construction of a binary mask. In order to generate CP, regions of interest, i.e. areas with skin coloured pixels, have to be identified in each image from a given set of training images. To manually create a binary mask which separates the skin from the background is a laborious task, especially for reasonably large image databases. In order to automating the processing, the training images can be segmented using the maximum and minimum thresholds for a logarithmic hue [11]. Comparing to the conventional angular hue, the logarithmic one is more robust to image deviations caused by varying image acquisition conditions, e.g. illumination. Figure 3.4 depicts an image segmented with the six different sets of the hue thresholds. As illustrated by these six examples, the changing thresholds results in different percentages of background and foreground (hand) pixels in the thresholded image.

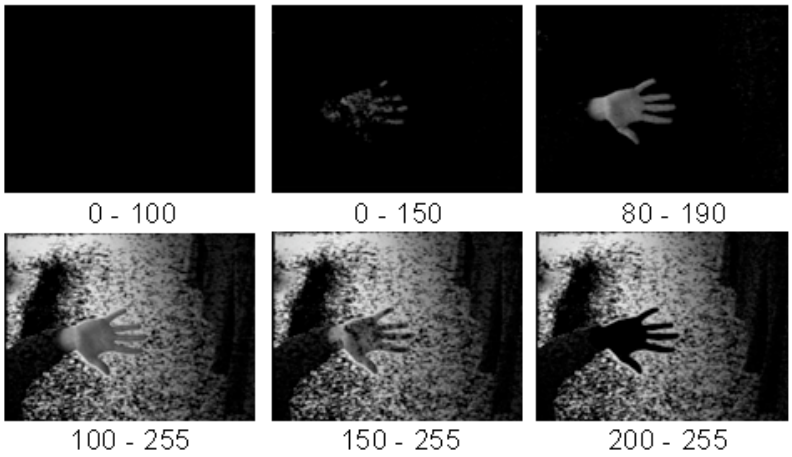


Figure 3.4: Segmentation with a range of thresholds.

Optimal thresholds can be derived by a trial-and-error procedure taking into consideration that the hand must be extracted from the image with the minimum amount of the background pixels as well as these thresholds must be flexible enough to be applied to a variety of similar images in the training set. However, a simple logic suggests that regardless of the chosen threshold pair, no range is specific enough to accurately extract a hand from its background, yet at the same time is general enough to satisfactorily accommodate inter-image variations. One possible solution to this problem is to apply to the training images a relatively general threshold set removing most but not all of the background noise and perform additional processing, e.g. median filtering of the hue image component and a simple morphological opening, to increase the accuracy of segmentation. These latter

two operations remove a large percentage of the background noise, as well as fill holes and gaps on the contour of the hand silhouette. Finally, the largest connected region of skin coloured pixels is extracted from an image through the use of the Haralick-Shapiro connected component algorithm described in Section 4 (see also [7, 8]. This processing strategy assumes other connected regions of the skin colour are generally much smaller than the hand region. The remaining non-background pixels are considered as the skin regions and used as the binary mask for the training of the predicate. Selection of results before and after applying the median filter and the morphological operator are depicted in Figs 3.5 and 3.6.

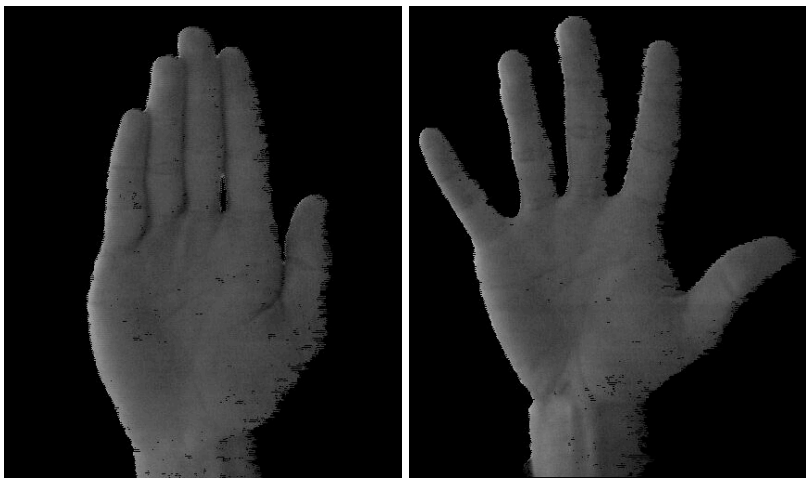


Figure 3.5: Results after hue thresholding.

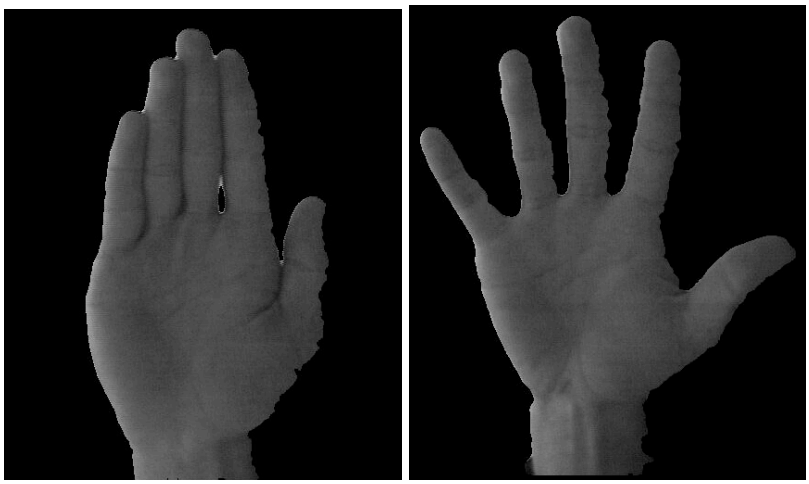


Figure 3.6: Post-processing results (morphological opening and median filtering).

Training process in Fig. 3.7. The hue and saturation pair (H,S) of each pixel is used to index the colour predicate. Pixels with too large and too small intensity are discarded because both the hue and saturation become unreliable in those ranges [10]. For this reason, the process in Fig. 3.7 removes 5% of pixels at either end of the intensity interval $[0, 255]$.

A broad positively weighted Gaussian window is then used to increment the neighbourhood of each indexed skin pixel, whereas a narrower negatively weighted window decrements the neighbourhood of the non-skin pixels in the predicate. As described in [10], the Gaussian smoothing and inclusion of the negatively valued background pixels to CP considerably improves the segmentation results.

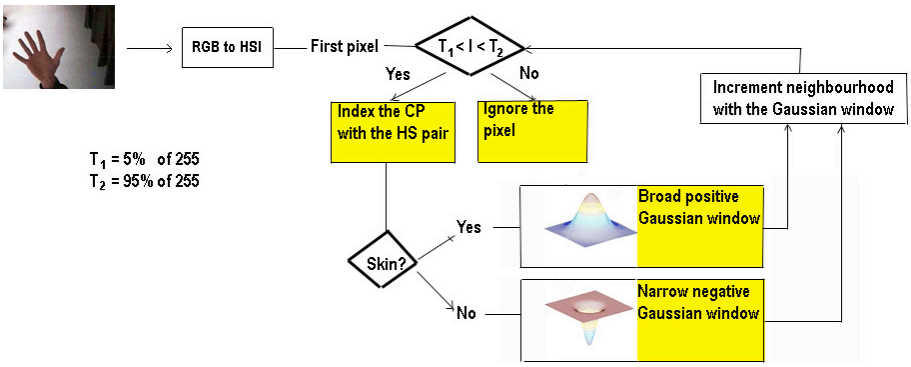


Figure 3.7: Training of the colour predicate

Once all the training images have been used to increment the colour predicate, the resulting trained CP is a Hue-Saturation plane, or table consisting of thef positive and negative H-S pairs. The predicate can then be separated into the skin and non-skin regions by thresholding this H-S plane, thus creating the binary predicate depicted in Fig. 3.8.

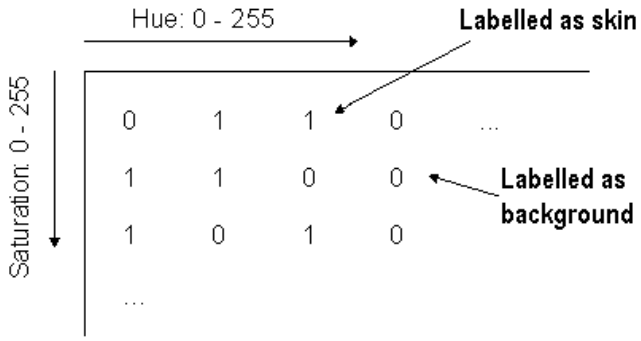


Figure 3.8: Binary colour predicate

Using the Colour Predicate. Once the predicate has been trained, it can be used to identify regions of skin within an image by simply using the CP index corresponding to a Hue-Saturation pair in each pixel. The purpose of the CP approach is not to obtain a higher quality segmentation but rather to provide means by which a large number of different input images from various users and environmental conditions can be automatically segmented to a reasonable quality while requiring a minimum amount of processing time. This is evident from Fig. 3.9 showing selected segmented images. Each silhouette exhibits small gaps and contour irregularities, yet it is of sufficiently high quality for pose classification and its complete processing takes on the average just 94 ms.

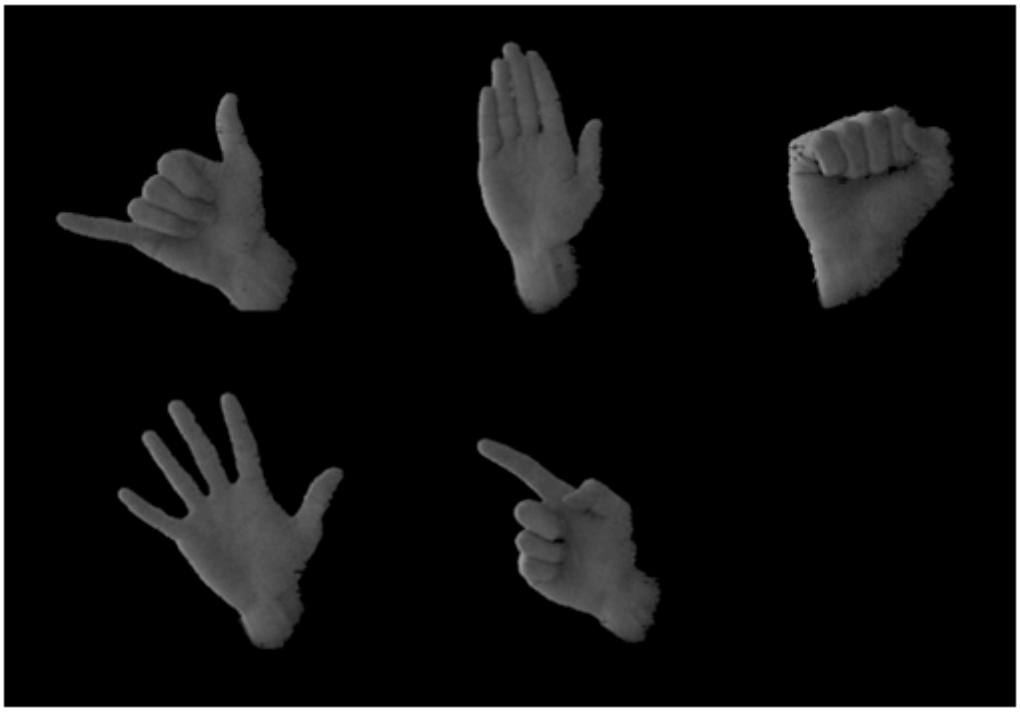


Figure 3.9: Image segmentation with a trained CP.

Chapter 4

Binary Machine Vision

In many practical cases, image analysis is simplified and accelerated by converting initial greyscale or colour images into binary (black – white or object – background) images. The simplest binarisation of a greyscale image is performed by comparing signals to a fixed or adaptive threshold.

4.1 Thresholding greyscale Images

The question of thresholding is how to automatically determine the threshold value. Since the threshold value separates the dark background from the bright object (or vice versa), the separation could ideally be done if the probability distributions of dark pixels and of bright pixels are known. Such a threshold value might equalise the probability of the two kinds of errors: of assigning a background pixel to a binary object and of assigning an object pixel to a binary background. More complex thresholding techniques use a spatially varying threshold to compensate for a variety of local spatial context effects (such a spatially varying threshold can be thought as a background normalisation).

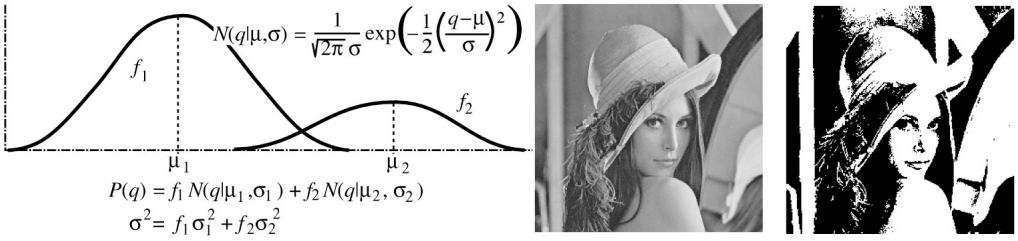
The independent distributions of dark and bright pixels are usually unknown, so that a normalised grey level histogram (GLH) $\mathbf{F}(\mathbf{g}) = [F(q|\mathbf{g}) : \ q = 0, ..., q_{\max}]$. for a given greyscale image $\mathbf{g}$, called also an empirical marginasl grey level distribution is used for finding a threshold. Each GLH component $F(q|\mathbf{g})$ is a relative number of pixels with a grey level q in the image $\mathbf{g}$. Most popular methods for thresholding are (*i*) minimisation of the within-group grey level variance (Otsu algorithm, 1979) and (*ii*) approximation of the GLH by a mixture of two Gaussian distributions (Kittler–Illingworth algorithm, 1985).

Minimising the within-group variance. For a threshold $0 \leq t \leq q_{\max}$, the within-group variance is equal to $s_t^2 = f_{1,t}s_{1,t}^2 + f_2(t)s_{2,t}^2$ where $f_{a,t}$ and $s_{a,t}^2$ denote the relative frequency and the variance of grey levels in a group $a = 1, 2$, respectively. Here, $f_{1,t} = \sum_{q=0}^{t} F(q|\mathbf{g})$; $f_{2,t} = \sum_{q=t+1}^{q_{\max}} F(q|\mathbf{g})$; $s_{1,t}^2 = \sum_{q=0}^{t}(q - m_{1,t})^2 F(q|\mathbf{g})$; $s_{2,t}^2 = \sum_{q=t+1}^{q_{\max}}(q -$

$m_{2,t})^2 F(q|\mathbf{g})$ where $m_{1,t} = \sum_{q=0}^{t} q\ F(q|\mathbf{g})$ and $m_{2,t} = \sum_{q=t+1}^{q_{\max}} q\ F(q|\mathbf{g})$. The chosen threshold $t^* = \arg\min_t s_t^2$. If the fractions of pixels in each mode are far from being approximately equal, this will not necessarily produce the correct answer.

Minimising the Kullback information distance. A mixture of two Gaussian distributions with parameters $\mathbf{m} = (f_1, \mu_1, \sigma_1, f_2, \mu_2, \sigma_2)$ is specified as $P(q) = f_1 N(q|\mu_1, \sigma_1) + f_2 N(q|\mu_2, \sigma_2)$ where $N(q|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-0.5\left(\frac{q-\mu}{\sigma}\right)^2\right)$. Approximation of the grey level histogram with the Gaussian mixture results in thresholding by minimizing the Kullback information distance $J(F(\mathbf{g};\mathbf{P}|\mathbf{m}) = \sum_{q=0}^{q_{\max}} P(q) \log \frac{P(q)}{F(q|\mathbf{g})}$ (Kittler–Illingworth, 1985). The desired threshold should minimise only the second term: $J_2(F(\mathbf{g};\mathbf{P}|\mathbf{m})) = -\sum_{q=0}^{q_{\max}} P(q) \log F(q|\mathbf{g})$. Simplifying assumption to compute the threshold is that the modes are well separated so that $F(q|\mathbf{g}) \approx f_1 N(q|\mu_1, \sigma_1)$ if $q \le t$ and $f_2 N(q|\mu_2, \sigma_2)$ if $q > t$. Under this assumption, the mixture parameters are easily estimated from the GLH.

Using the estimates $f_{1,t}, ..., s_{2,t}$, the information measure $J_2 = -f_1 \log f_1 + 0.5(f_1 \log \sigma_1^2 + f_2 \log \sigma_2^2)$ can be evaluated for each threshold t, and the value that minimises J_2 is then the best threshold. Cho, Haralick, and Yi in 1989 (see [8]) improved this technique to maximise the probability of correct classification. The corrected parameter values take account of the truncated distributions.



Signal clustering with EM algorithm. The above thresholding is a particular case of more general problem of finding two or more clusters of "similar" signals in a given large set of signals. The similarity is assumed to account for only the signal values but not for other image properties (such as adjacency in the image plane or so on).

Let $\mathbf{Q} = \{q_i :\ i = 1, \ldots, n\}$ be a signal set to split into $K \ge 2$ clusters. A very special case of the general EM (Expectation – Maximisation) algorithm forms clusters that maximise the likelihood of the signals under the following conditions: (*i*) signals of each cluster have a probability distribution of known type but with *a priori* unknown parameters and (*ii*) signals in the whole set are statistically independent. Let $f(q;\theta_k)$ denote the probability distribution function or the probability density function for the cluster k. Let $\mathbf{p} = (p_k :\ k = 1, \ldots, K)$ be prior probabilities of each cluster for each particular signal: $\sum_{k=1}^{K} p_k = 1$. Then the probability (or probability density) of the signal q_i is given by the

mixture $\Pr(q_i) = \sum_{k=1}^{K} p_k f(q_i;\theta_k)$. Due to the assumed signal independence, the overall probability of the signal set is $\Pr(\mathbf{Q}) = \prod_{i=1}^{n} \Pr(q_i) \equiv \prod_{i=1}^{n} \sum_{k=1}^{K} p_k f(q_i;\theta_k)$. The log-likelihood function is $L(\Theta,\mathbf{p}) = \log \Pr(\mathbf{Q})$, therefore, the maximum likelihood clustering $(\Theta^*,\mathbf{p}^*) = \arg\max_{\Theta,\mathbf{p}} \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} p_k f(q_i;\theta_k)\right)$ is the conditional optimisation under the unit sum of the prior cluster probabilities.

The conditional maximisation results in the following systems of equations obtained by setting to zero partial derivatives of the Lagrange function $L(\Theta,\mathbf{p}) - \lambda\left(\sum_{k=1}^{K} p_k - 1\right)$ with respect to the cluster parameters: $\forall k = 1,\ldots,K$

$$\frac{\partial L(\Theta,\mathbf{p})}{\partial p_k} = \sum_{i=1}^{n} \frac{f(q_i;\theta_k)}{\sum_{\kappa=1}^{K} p_\kappa f(q_i;\theta_\kappa)} \tag{4.1}$$

$$\frac{\partial L(\Theta,\mathbf{p})}{\partial \theta_k} = \sum_{i=1}^{n} \frac{p_k \frac{\partial}{\partial\theta_k} f(q_i;\theta_k)}{\sum_{\kappa=1}^{K} p_\kappa f(q_i;\theta_\kappa)} \tag{4.2}$$

Because of complexity of this system, the EM algorithm provides an iterative search for a local maximum of the likelihood. Let $\pi(k|q_i)$ for $k = 1,\ldots,K$ and $i = 1,\ldots,n$ denote the weight that can be considered as an analogue of *a posteriori* probability of the claster k for the signal q_i:

$$\pi(k|q_i) = \frac{p_k f(q_i;\theta_k)}{\sum_{\kappa=1}^{K} p_\kappa f(q_i;\theta_\kappa)}$$

For all $i = 1,\ldots,n$, it holds that $\pi(k|q_i) \geq 0$ and $\sum_{k=1}^{K} \pi(k|q_i) = 1$. By using the above weights, the log-likelihood $L(\Theta,\mathbf{p})$ can be rewritten as

$$L(\Theta,\mathbf{p}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \pi(k|q_i)\log p_k + \sum_{i=1}^{n}\sum_{k=1}^{K} \pi(k|q_i)\log f(q_i;\theta_k) - \sum_{i=1}^{n}\sum_{k=1}^{K} \pi(k|q_i)\log \pi(k|q_i)$$

Then the first sum in the above formula depends only on the desired priors $\mathbf{p}$, and the second sum depends only on the desired cluster parameters θ_k; $k = 1,\ldots,K$. The iterative EM-like parameter estimation consists at each iteration of two successive steps, the first step assuming that all the weights $\pi(k|q_i)$ are fixed and the second step assuming that the priors and cluster parameters are fixed.

The **first step** obtains the current values of the priors and cluster parameters by conditional maximisation of the likelihood. Under the fixed weights $\pi(k|q_i)$, the priors that conditionally maximise the first sum under their unit sum constraint are as follows: $p_k^* = \frac{1}{n}\sum_{i=1}^{n} \pi(k|q_i)$. The cluster parameters Θ are obtained by the unconditional maximisation of the second sum:

$$\forall k = 1,\ldots,K \quad \sum_{i=1}^{n} \pi(k|q_i)\, \frac{\frac{\partial}{\partial\theta_k} f(q_i;\theta_k)}{f(q_i;\theta_k)} = 0$$

The **second step** returns the weights $\pi(k|q_i)$ which provide the conditional maximum of the likelihood under the above conditions on these weights (i.e. the un it sum for each cluster k):

$$\frac{\partial}{\partial \pi(k|q_i)} L(\Theta, \mathbf{p}) = \log p_k + \log f(q_i;\theta_k) - \log \pi(k|q_i) - 1 - \lambda_i \quad \text{so that} \quad (4.3)$$

$$\pi(k|q_i) = p_k f(q_i;\theta_k) \exp(1+\lambda_i) \quad (4.4)$$

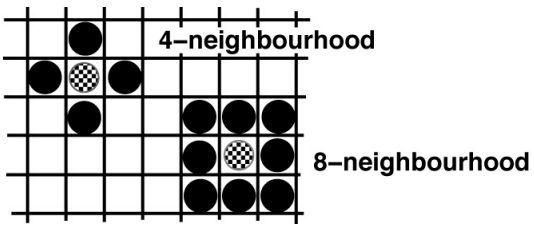
This results just in the above-mentioned relationships for the weights.

Iterative maximisation of the likelihood by successive repetition of these two steps is performed until a particular stopping criterion is satisfied, for instance, until a relative change of the likelihood value $L(\Theta_t, \mathbf{p}_t)$ at iteration t with respect to the previous value $L(\Theta_{t-1}, \mathbf{p}_{t-1})$ becomes less than a given threshold.

In a particular case of normal (Gaussian) distributions of signals in each cluster, the cluster parameter θ_k consists of the mean signal μ_k and variance σ_k^2. It is easy to show that in this case the parameter estimation algorithm becomes as follows: (1) **initialisation**: select the number of clusters K and the parameters $\Theta_0 = \left((\mu_{k,0}, \sigma_{k,0}^2) : \ k = 1, \ldots, K\right)$ and priors $\mathbf{p}_0 = \left(p_{k,0} = \frac{1}{K} : \ k = 1, \ldots, K\right)$ of the initial clusters and (2) **iterative maximisation**: at each iteration $t = 1, 2, \ldots$, compute first the current weights $\pi_t(k|q_i)$ using the previous cluster priors $\mathbf{p}_{t-1}$ and parameters Θ_{t-1} and then re-estimate both the priors and parameters using the relationships for the Gaussian case, i.e. $p_{k,t} = \frac{1}{n}\nu_{k,t}$; $\mu_{k,t} = \frac{1}{\nu_{k,t}} \sum_{i=1}^{n} \pi_t(k|q_i) q_i$, and $\sigma_{k,t}^2 = \frac{1}{\nu_{k,t}} \sum_{i=1}^{n} \pi_t(k|q_i) q_i^2 - \mu_{k,t}^2$ where $\nu_{k,t} = \sum_{i=1}^{n} \pi_t(k|q_i)$. This variant of the EM algorithm has sometimes rather slow convergence to the local maximum close to the initial point in the parameter space. But in many clustering problems it outperforms heuristic counterparts. After signal clustering, an image can be converted into a binary one by selecting one particular cluster as an object and using all other clusters as a background. The signal clusters are obtained by assigning every signal q_i to the cluster with the maximum posterior probability $\pi_T(k|q_i)$ at the last iteration T.

4.2 Connected regions in binary images

Pixel $\mathbf{p}_i$ in a region R is **connected** to $\{\mathbf{p}_j$ if there is a sequence $\{\mathbf{p}_i, ..., \mathbf{p}_j\}$ such that each two successive pixels $\{\mathbf{p}_k, \mathbf{p}_{k+1}\}$ are the nearest neighbours and all the pixels are in R. The region R is a **connected region** if each pair of pixels in it is connected. Definitions of 4- and 8-neighbourhood of a pixel are given below:



Iterative segmentation of a binary image. An iterative algorithm proposed by Haralick (1981) conducts a sequence of top–down label propagation followed by bottom–up label propagation iterated until no label changes occur. The minimum label of the neighbours is selected to assign to each pixel (see Fig. 4.1).

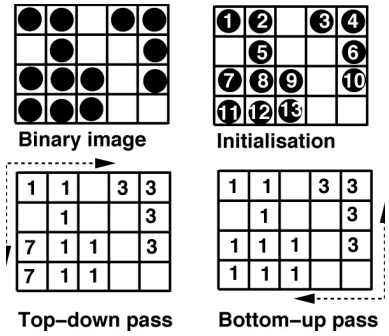
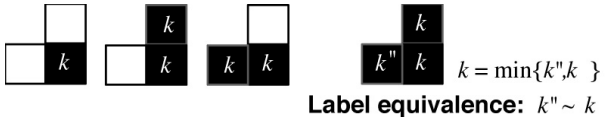


Figure 4.1: Iterative segmentation.

Connected components are labelled as follows. At the **initialisation stage** the initial label is set to zero $k = 0$ and the region map **l** is set to zero: $l(x, y) = 0$ for all image rows y and columns x. Then the **sequential components labeling** is performed along the top–to–bottom and left–to–right scan:

- If $g(x, y) = 1$ **AND** ($g(x, y-1) = 0$ **AND** $g(x-1, y) = 0$), start the new region: $k := k + 1$; $l(r.y) = k$.
- If $g(x, y) = 1$ **AND** ($g(x, y-1) = 1$ **AND** $g(x-1, y) = 0$), continue the previous region: $l(x, y) = l(x, y-1)$.
- If $g(x, y) = 1$ **AND** ($g(x, y-1) = 0$ **AND** $g(x-1, y) = 1$), continue the previous region: $l(x, y) = l(x-1, y)$.
- If $g(x, y) = 1$ **AND** ($g(x, y-1) = 1$ **AND** $g(x-1, y) = 1$), merge the previous regions: $l(x, y) = \min\{l(x-1, y), l(x, y-1)\}$.



Then the **relabelling** is performed by using the label equivalences. It can be implemented by a depth-first search in a graph structure defined by the set of equivalences: the nodes of the graph are region labels, and the edges are pairs of labels which are equivalent. In so doing, connected components of the graph structure defined by the set of label equivalences collected during the sequential component labelling are found and each connected set of the equivalent labels is relabelled.

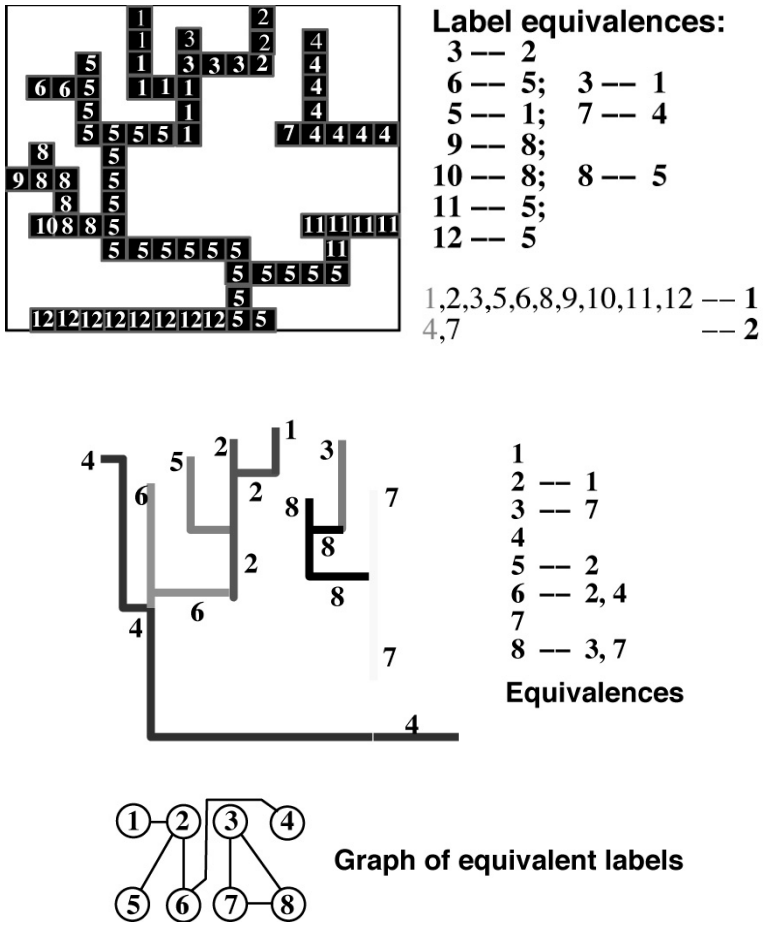


Figure 4.2: One more example of the connected components labelling.

Space-efficient two-pass connected component labelling The main problem of the classical connected components labelling is the global equivalence table: for large images with many regions, the equivalence table can become very large. One solution to the space problem is the use of a small local equivalence table that stores only the equivalences detected from the current image line and the line that precedes it. Thus the maximum number of equivalences is the number of pixels per line. These equivalences are then used in the propagation step to the next line.

Not all the equivalencing is done by the end of the first top–down pass, and the second pass bottom–up is required both to find the remainder of the equivalences and assign the final labels.

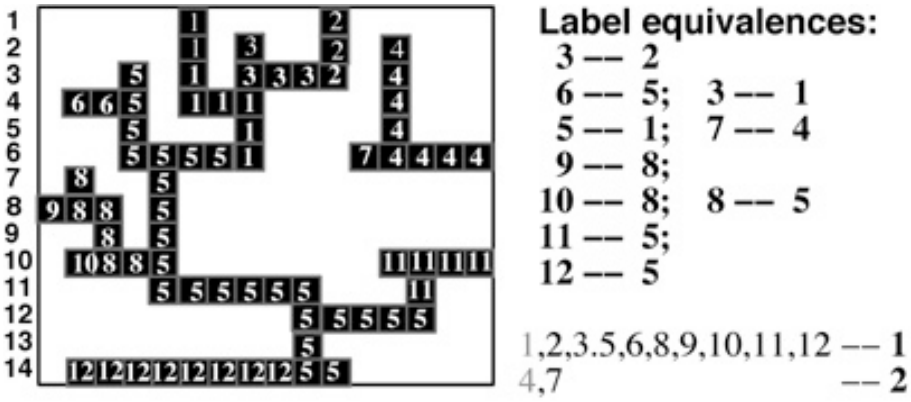


Figure 4.3: Space-efficient labeling:label equivalences.

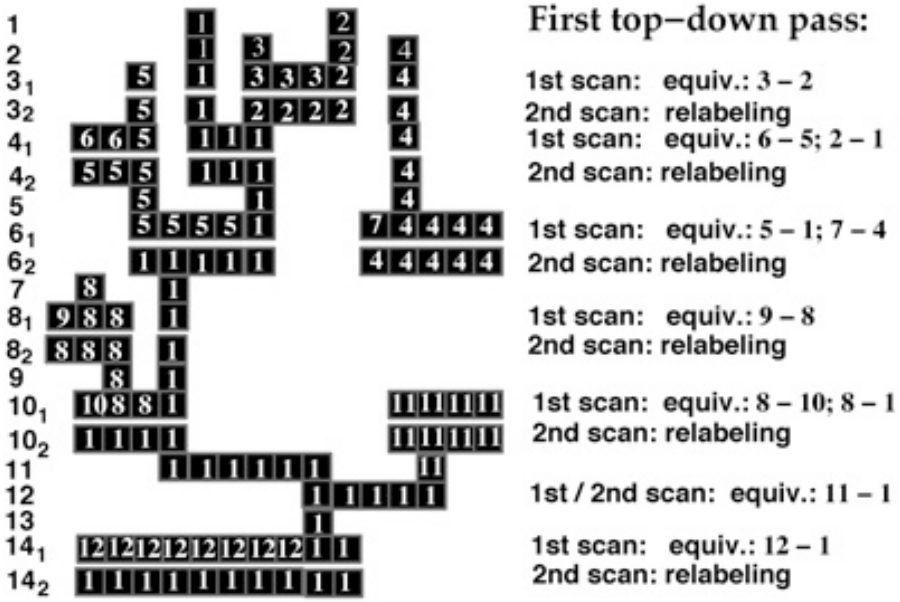


Figure 4.4: Space-efficient labeling: first top-down pass.

Appendix A

More on Camera Calibration

This appendix addresses one more solution of the problem of calibrating a single camera using a given set of 3D ground control points and corresponding image points. Analytical and numerical approaches for approximating the desired camera model parameters are discussed.

A.1 Initial calibration of a single camera

We consider a simplified camera model given by a fundamental perspective projection equation [8]. This simplification takes no account of additional non-linear image distortions caused by the lens system as in the more general Tsai's calibration. Let (x, y) and (X, Y, Z) be, respectively, 2D coordinates of image points and 3D coordinates of scene points, $\mathbf{R}(\omega, \phi, \kappa) = \|r_{kl}\|_{k,l=1}^{3}$ denote a 3D rotation matrix of the camera reference frame with respect to the world one, s_x and s_y are scale factors for the image, and f denote a camera constant. The rotation matrix depends on three rotation angles around axes X (tilt ω), Y (pan ϕ), and Z (swing κ) as follows:

$$\mathbf{R}(\omega, \phi, \kappa) \;=\; \mathbf{R}_Z(\kappa)\mathbf{R}_Y(\phi)\mathbf{R}_X(\omega) \;=$$
$$\left\| \begin{array}{ccc} \cos\phi\cos\kappa & \sin\omega\sin\phi\cos\kappa \;+\; \cos\omega\sin\kappa & -\cos\omega\sin\phi\cos\kappa \;+\; \sin\omega\sin\kappa \\ -\cos\phi\sin\kappa & -\sin\omega\sin\phi\sin\kappa \;+\; \cos\omega\cos\kappa & \cos\omega\sin\phi\sin\kappa \;+\; \sin\omega\cos\kappa \\ \sin\phi & -\sin\omega\cos\phi & \cos\omega\cos\phi \end{array} \right\| \tag{A.1}$$

Then the 3D scene and 2D image points are related by:

$$\begin{aligned} x \;&=\; x_0 \;+\; f \cdot s_x \cdot \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \;; \\ y \;&=\; y_0 \;+\; f \cdot s_y \cdot \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \end{aligned} \tag{A.2}$$

where (X_0, Y_0, Z_0) denotes the camera position in the world reference frame and (x_0, y_0)

is the principal point of the image.

Usually the scale factors s_x and s_y are close to unity so that in initial stage of camera parameter estimation one can omit them: $s_x \simeq s_y \simeq 1$. To obtain a close-enough raw estimates of the other 9 parameters $\mathbf{p} = (x_0, y_0, f, X_0, Y_0, Z_0, \omega, \phi, \kappa)$, given a set of corresponding scene and image points

$$\{(x_n, y_n),\ (X_n, Y_n, Z_n) :\ n = 1, \ldots, N\},$$

we reduce the estimation to a weighted least-squares problem:

$$\begin{array}{c} \min\limits_{\mathbf{a}} \{\mathbf{a}^t \mathbf{A} \mathbf{a}\}; \\ \mathbf{a}^t \mathbf{C} \mathbf{a} = \text{const.} \end{array} \tag{A.3}$$

Here and below, t indicates a transposition, $\mathbf{A}$ denotes a 12×12 matrix of sums of the coordinate products, $\mathbf{C}$ denotes a 12×12 matrix of the constraints, and $\mathbf{a}$ is 12×1 vector with components depending on the desired calibration parameters.

To formulate the problem in Eq. (A.3), the relations of Eq. (A.2) are represented by:

$$x = \frac{\mathbf{a}_{1,4}^t \mathbf{X}}{\mathbf{a}_{9,12}^t \mathbf{X}}\ ; \quad y = \frac{\mathbf{a}_{5,8}^t \mathbf{X}}{\mathbf{a}_{9,12}^t \mathbf{X}} \tag{A.4}$$

where $\mathbf{X}$ is a 4×1 vector of homogeneous 3D point coordinates: $\mathbf{X}^t = (X, Y, Z, 1)$ and three vectors 4×1: $\mathbf{a}_{1,4}^t, \mathbf{a}_{5,8}^t, \mathbf{a}_{9,12}^t$ are consecutive parts of the vector $\mathbf{a}$ as follows: $\mathbf{a}^t = \left(\mathbf{a}_{1,4}^t, \mathbf{a}_{5,8}^t, \mathbf{a}_{9,12}^t\right)$. Their indices indicate the successive quadruples of the components of the vector $\mathbf{a}$ (in particular, $\mathbf{a}_{14}^t = (a_1, a_2, a_3, a_4)$). These components are expressed in terms of the camera calibration parameters in Eq. (A.2), as follows:

$$\begin{array}{rcl}
a_1 &=& r_{11} f + r_{31} x_0; \\
a_2 &=& r_{12} f + r_{32} x_0; \\
a_3 &=& r_{13} f + r_{33} x_0; \\
a_4 &=& -[(r_{11} X_0 + r_{12} Y_0 + r_{13} Z_0)\, f \ +\ (r_{31} X_0 + r_{32} Y_0 + r_{33} Z_0)\, x_0]; \\
a_5 &=& r_{21} f + r_{31} y_0; \\
a_6 &=& r_{22} f + r_{32} y_0; \\
a_7 &=& r_{23} f + r_{33} y_0; \\
a_8 &=& -[(r_{21} X_0 + r_{22} Y_0 + r_{23} Z_0)\, f \ +\ (r_{31} X_0 + r_{32} Y_0 + r_{33} Z_0)\, y_0]; \\
a_9 &=& r_{31}; \\
a_{10} &=& r_{32}; \\
a_{11} &=& r_{33}; \\
a_{12} &=& -\left(r_{31} X_0 + r_{32} Y_0 + r_{33} Z_0\right).
\end{array} \tag{A.5}$$

The least-square problem in Eq. (A.3) is obtained by replacing the non-linear minimization problem:

$$\min_{\mathbf{p}} \sum_{n=1}^{N} \left(x_n - \frac{\mathbf{a}_{1,4}^t \mathbf{X}_n}{\mathbf{a}_{9,12}^t \mathbf{X}_n} \right)^2 + \left(y_n - \frac{\mathbf{a}_{5,8}^t \mathbf{X}_n}{\mathbf{a}_{9,12}^t \mathbf{X}_n} \right)^2 \tag{A.6}$$

with the following apoproximate one:

$$\min_{\mathbf{p}} \sum_{n=1}^{N} \left(x_n \mathbf{a}_{9,12}^t \mathbf{X}_n - \mathbf{a}_{1,4}^t \mathbf{X}_n \right)^2 + \left(y_n \mathbf{a}_{5,8}^t \mathbf{X}_n - \mathbf{a}_{9,12}^t \mathbf{X}_n \right)^2. \tag{A.7}$$

Therefore, the matrix $\mathbf{A}$ in Eq. (A.3) has a following structure:

$$\mathbf{A} = \left\| \begin{array}{ccc} \mathbf{W} & \mathbf{0} & -\mathbf{V}_x \\ \mathbf{0} & \mathbf{W} & -\mathbf{V}_y \\ -\mathbf{V}_x & -\mathbf{V}_y & \mathbf{V}_{xy} \end{array} \right\| \tag{A.8}$$

where

$$\mathbf{W} = \left\| \begin{array}{cccc} S_{XX} & S_{XY} & S_{XZ} & S_X \\ S_{XY} & S_{YY} & S_{YZ} & S_Y \\ S_{XZ} & S_{YZ} & S_{ZZ} & S_Z \\ S_X & S_Y & S_Z & n \end{array} \right\|; \tag{A.9}$$

$$\mathbf{V}_x = \left\| \begin{array}{cccc} S_{xXX} & S_{xXY} & S_{xXZ} & S_{xX} \\ S_{xXY} & S_{xYY} & S_{xYZ} & S_{xY} \\ S_{xXZ} & S_{xYZ} & S_{xZZ} & S_{xZ} \\ S_{xX} & S_{xY} & S_{xZ} & S_x \end{array} \right\|; \tag{A.10}$$

$$\mathbf{V}_y = \left\| \begin{array}{cccc} S_{yXX} & S_{yXY} & S_{yXZ} & S_{yX} \\ S_{yXY} & S_{yYY} & S_{yYZ} & S_{yY} \\ S_{yXZ} & S_{yYZ} & S_{yZZ} & S_{yZ} \\ S_{yX} & S_{yY} & S_{yZ} & S_y \end{array} \right\|; \tag{A.11}$$

$$\mathbf{V} = \left\| \begin{array}{cccc} S_{(xx+yy)XX} & S_{(xx+yy)XY} & S_{(xx+yy)XZ} & S_{(xx+yy)X} \\ S_{(xx+yy)XY} & S_{(xx+yy)YY} & S_{(xx+yy)YZ} & S_{(xx+yy)Y} \\ S_{(xx+yy)XZ} & S_{(xx+yy)YZ} & S_{(xx+yy)ZZ} & S_{(xx+yy)Z} \\ S_{(xx+yy)X} & S_{(xx+yy)Y} & S_{(xx+yy)Z} & S_{xx+yy} \end{array} \right\|. \tag{A.12}$$

Here, $S_{\dots}$ denote sums of the products of the point 3D and 2D coordinates, for instance, $S_{XX} = \sum_{n=1}^{N} X_n^2;\ S_{xYZ} = \sum_{n=1}^{N} x_n Y_n Z_n;\ S_{(xx+yy)XZ} = \sum_{n=1}^{N} (x_n^2 + y_n^2) X_n Z_n$, etc.

The 2D coordinates in Eq. (A.2) are invariant to a scale of the 3D coordinates of the points. The above relations in Eq. (A.5) between the parameters to be estimated and components of the vector $\mathbf{a}$ show that these latter components have an obvious constraint $a_9^2 + a_{10}^2 + a_{11}^2 = 1$ which follows from the particular form of the rotation matrix $\mathbf{R}$ of Eq. (A.1). Thus, the constraint matrix in Eq. (A.3) is a diagonal one with three non-zero diagonal unit components which cut out only the components $a_9, \dots, a_{11}$ of the vector $\mathbf{a}$.

Estimation of the model parameters. Now, the optimization problem of Eq. (A.3) is as follows:

$$\begin{array}{rcl} \mathbf{a}^*_{9,12} & = & \arg\min\limits_{\mathbf{a}_{9,12}} \left\{ \mathbf{a}^t_{9,12} \mathbf{U} \mathbf{a}_{9,12} \right\}; \\ a_9^2 + a_{10}^2 + a_{11}^2 & = & 1. \end{array} \tag{A.13}$$

Here, the matrix $\mathbf{U}$ has the following form:

$$\mathbf{U} = \mathbf{V}_{xy} - \mathbf{V}_x \mathbf{W}^{-1} \mathbf{V}_x - \mathbf{V}_y \mathbf{W}^{-1} \mathbf{V}_y.$$

The desired vector $\mathbf{a}^*_{9,12}$ in Eq. A.13 can be obtained as the eigenvector of the matrix $\mathbf{U}$ which has the minimal eigenvalue under the involved constraints. To be more precise, the above normalization of the components $a^*_9,\ a^*_{10}, a^*_{11}$, that is, $(a^*_9)^2 + (a^*_{10})^2 + (a^*_{11})^2 = 1$ is applied to the initial eigenvectors with corresponding changing of the initial eigenvalues. The resulting vector allows to compute both other vectors $\mathbf{a}^*_{1,4} = \mathbf{W}^{-1}\mathbf{V}_x \mathbf{a}^*_{9,12}$ and $\mathbf{a}^*_{5,8} = \mathbf{W}^{-1}\mathbf{V}_y \mathbf{a}^*_{9,12}$ to form the desired solution $\mathbf{a}^*$ of the optimizing problem in Eq. (A.3).

We omit below, for simplicity, the index $(^*)$ of the components of this optimal solution. It allows us to compute the raw estimates of the camera parameters $\mathbf{p}$. First two camera orientation angles ϕ and ω (see Eq. (A.1)) are obtained from the components a_9, a_{10}, a_{11}. The pan angle ϕ has two possible values:

$$\phi = \arctan\left(\frac{a_9}{\sqrt{1 - a_9^2}} \right) \text{ or } \phi = \arctan\left(\frac{a_9}{-\sqrt{1 - a_9^2}} \right) \tag{A.14}$$

so that we obtain two possible final sets of the angles giving the same orientation matrix $\mathbf{R}$. The tilt angle ω depends on the pan angle value as follows:

$$\omega = \arctan\left(\frac{-a_{10} \cdot \text{sign}(\cos\phi)}{a_{11} \cdot \text{sign}(\cos\phi)} \right). \tag{A.15}$$

It can be easily shown that the 3D camera position is next obtained by solving the following linear equation system:

$$\begin{array}{rcl} a_1 X_0 + a_2 Y_0 + a_3 Z_0 & = & -a_4; \\ a_5 X_0 + a_6 Y_0 + a_7 Z_0 & = & -a_8; \\ a_9 X_0 + a_{10} Y_0 + a_{11} Z_0 & = & -a_{12}; \end{array} \tag{A.16}$$

so that the desired solution is as follows:

$$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} = - \begin{Vmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \end{Vmatrix}^{-1} \begin{pmatrix} a_4 \\ a_8 \\ a_{12} \end{pmatrix}. \tag{A.17}$$

Now, by using the estimated angles ϕ and ω, we form the following system of linear equations for getting the swing angle κ, the camera constant f, and the principal point (x_0, y_0):

$$\left\| \begin{array}{cccc} \cos\phi & 0 & \sin\phi & 0 \\ \sin\omega\sin\phi & \cos\omega & -\sin\omega\cos\phi & 0 \\ -\cos\omega\sin\phi & \sin\omega & \cos\omega\cos\phi & 0 \\ 0 & -\cos\phi & 0 & \sin\phi \\ \cos\omega & -\sin\omega\sin\phi & 0 & -\sin\omega\cos\phi \\ \sin\omega & \cos\omega\sin\phi & 0 & \cos\omega\cos\phi \end{array} \right\| \left(\begin{array}{c} f\cos\kappa \\ f\sin\kappa \\ x_0 \\ y_0 \end{array} \right) = \left(\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_5 \\ a_6 \\ a_7 \end{array} \right) \quad \text{(A.18)}$$

This overconstrained system can be solved in a least-square way so that the desired estimates are then as follows:

$$\begin{array}{rcl} f\cos\kappa \equiv f_c & = & 0.5\left(a_1\cos\phi + a_2\sin\omega\sin\phi - a_3\cos\omega\sin\phi + a_6\cos\omega + a_7\sin\omega\right); \\ f\sin\kappa \equiv f_s & = & 0.5\left(a_2\cos\omega + a_3\sin\omega - a_5\cos\phi - a_6\sin\omega\sin\phi + a_7\cos\omega\sin\phi\right); \\ x_0 & = & a_1\sin\phi - a_2\sin\omega\cos\phi + a_3\cos\omega\cos\phi; \\ y_0 & = & a_5\sin\phi - a_6\sin\omega\cos\phi + a_7\cos\omega\cos\phi. \end{array} \quad \text{(A.19)}$$

Therefore, $f = \sqrt{f_c^2 + f_s^2}$ and $\kappa = \arctan\left(\frac{f_s}{f_c}\right)$. The parameters

$$\mathbf{p}^t = (x_0, y_0, f, X_0, Y_0, Z_0, \omega, \phi, \kappa)$$

found can be then refined by a non-linear least-square technique such as that of [8].

A.2 Refinement of the calibration

Starting from the initial raw approximation, the camera model parameters can be obtained with higher precision by an iterative solution of the linearized problem in Eq. (A.6). The linearization is taken around each current approximate solution $\mathbf{p}$.

Let us denote $e_{x,n}$ and $e_{y,n}$ the current x- and y-coordinate errors in Eq. (A.6), respectively:

$$\begin{array}{rcl} e_{x,n}(\mathbf{p}) & = & x_n - \tilde{x}_n; \\ e_{y,n}(\mathbf{p}) & = & y_n - \tilde{y}_n \end{array} \quad \text{(A.20)}$$

where the estimated coordinates $\tilde{x}_n$ and $\tilde{y}_n$ are given by Eq. (A.4):

$$\begin{array}{rcl} \tilde{x}_n & = & \frac{\mathbf{a}_{1,4}^t \mathbf{X}_n}{\mathbf{a}_{9,12}^t \mathbf{X}_n}; \\ \tilde{y}_n & = & \frac{\mathbf{a}_{5,8}^t \mathbf{X}_n}{\mathbf{a}_{9,12}^t \mathbf{X}_n}. \end{array} \quad \text{(A.21)}$$

The errors depend non-linearly on the parameters $\mathbf{p}$, and these non-linear transformations can be linearized around a current solution $\mathbf{p}$ by representing each estimated coordinate by a first-order Taylor series expansion:

$$\begin{array}{rcl} e_{x,n}(\mathbf{p}+\mathbf{\Delta p}) & = & e_{x,n}(\mathbf{p}) - \Delta\tilde{x}_n(\mathbf{p},\mathbf{\Delta p}); \\ e_{y,n}(\mathbf{p}+\mathbf{\Delta p}) & = & e_{y,n}(\mathbf{p}) - \Delta\tilde{y}_n(\mathbf{p},\mathbf{\Delta p}) \end{array} \tag{A.22}$$

where $\Delta\tilde{x}_n$ and $\Delta\tilde{y}_n$, the total derivatives of $\tilde{x}_n$ and $\tilde{y}_n$, respectively, are linear functions of the parameter adjustment vector $\mathbf{\Delta p}$:

$$\begin{array}{rcl} \Delta\tilde{x}_n(\mathbf{p},\mathbf{\Delta p}) & = & \mathbf{\Delta p}^t\mathbf{g}_{x,n}; \\ \Delta\tilde{y}_n(\mathbf{p},\mathbf{\Delta p}) & = & \mathbf{\Delta p}^t\mathbf{g}_{y,n} \end{array} \tag{A.23}$$

where and are the gradient vectors (that is, vectors of partial derivatives)

$$\begin{array}{rcl} \mathbf{g}_{x,n} & = & \left(\frac{\partial\tilde{x}_n}{\partial p_1}(\mathbf{p}^t), ..., \frac{\partial\tilde{x}_n}{\partial p_9}(\mathbf{p}^t)\right); \\ \mathbf{g}_{y,n} & = & \left(\frac{\partial\tilde{y}_n}{\partial p_1}(\mathbf{p}^t), ..., \frac{\partial\tilde{y}_n}{\partial p_9}(\mathbf{p}^t)\right). \end{array} \tag{A.24}$$

The total derivative shows the linear part of how the error will be perturbed if the parameter vector is perturbed by an amount $\mathbf{\Delta p}$.

When the linearized estimates of Eq. (A.23) are substituted into the least-square problem in Eq. (A.6), we obtain the following perturbed total square error:

$$\sum_{n=1}^{N} \left(\Delta e_{x,n}(\mathbf{p},\mathbf{\Delta p})\right)^2 + \left(\Delta e_{y,n}(\mathbf{p},\mathbf{\Delta p})\right)^2, \tag{A.25}$$

or

$$\sum_{n=1}^{N} \left(e_{x,n}(\mathbf{p}) - \mathbf{\Delta p}^t\mathbf{g}_{x,n}\right)^2 + \left(e_{y,n}(\mathbf{p}) - \mathbf{\Delta p}^t\mathbf{g}_{y,n}\right)^2, \tag{A.26}$$

and its minimization by $\mathbf{\Delta p}$ results in the linear system of equations as follows:

$$\sum_{n=1}^{N} \left(e_{x,n}(\mathbf{p}) - \mathbf{\Delta p}^t\mathbf{g}_{x,n}\right)\mathbf{g}_{x,n}^t + \left(e_{y,n}(\mathbf{p}) - \mathbf{\Delta p}^t\mathbf{g}_{y,n}\right)\mathbf{g}_{x,n}^t = 0. \tag{A.27}$$

This system can be easily represented in the matrix form:

$$\mathbf{d}(\mathbf{p}) - \mathbf{G}(\mathbf{p})\mathbf{\Delta p} = 0 \tag{A.28}$$

where the 9×9 symmetric matrix $\mathbf{G}(\mathbf{p})$ and 9×1 column vector $\mathbf{d}(\mathbf{p})$ are as follows:

$$\begin{array}{rcl} \mathbf{G}(\mathbf{p}) & = & \sum_{n=1}^{N} \mathbf{g}_{x,n}\mathbf{g}_{x,n}^t + \mathbf{g}_{y,n}\mathbf{g}_{y,n}^t; \\ \mathbf{d}(\mathbf{p}) & = & \sum_{n=1}^{N} e_{x,n}(\mathbf{p})\mathbf{g}_{x,n} + e_{y,n}(\mathbf{p})\mathbf{g}_{y,n}. \end{array} \tag{A.29}$$

When the matrix $\mathbf{G}(\mathbf{p})$ is non-singular, the system of Eq. (A.28) has a standard solution:

$$\mathbf{\Delta p} = \mathbf{G}^{-1}(\mathbf{p})\mathbf{d}(\mathbf{p}), \tag{A.30}$$

and the corrected parameters are as follows:

$$\mathbf{p}_{\text{corr}} = \mathbf{p} + \mathbf{\Delta p}. \tag{A.31}$$

If the raw estimates are in a sufficiently close vicinity of the optimal values, there should be about five – ten sequential corrections to reach the desired optimum [8].

Gradients of the coordinate estimates. The gradient components $g_{x,n,i} = \frac{\partial \tilde{x}_n}{\partial p_i}$ and $g_{y,n,i} = \frac{\partial \tilde{y}_n}{\partial p_i}$ in Eq. (A.24) to be used for refining the calibration parameters have the following obvious forms:

$$\begin{array}{rcl} g_{x,n,i} & = & \dfrac{\frac{\partial \mathbf{a}_{1,4}^t}{\partial p_i}\mathbf{X}_n}{\mathbf{a}_{9,12}^t\mathbf{X}_n} - \dfrac{\mathbf{a}_{1,4}^t\mathbf{X}_n\frac{\partial \mathbf{a}_{9,12}^t}{\partial p_i}\mathbf{X}_n}{\left(\mathbf{a}_{9,12}^t\mathbf{X}_n\right)^2}\ ; \\ g_{y,n,i} & = & \dfrac{\frac{\partial \mathbf{a}_{5,8}^t}{\partial p_i}\mathbf{X}_n}{\mathbf{a}_{9,12}^t\mathbf{X}_n} - \dfrac{\mathbf{a}_{5,8}^t\mathbf{X}_n\frac{\partial \mathbf{a}_{9,12}^t}{\partial p_i}\mathbf{X}_n}{\left(\mathbf{a}_{9,12}^t\mathbf{X}_n\right)^2}\ . \end{array} \tag{A.32}$$

The above dot products involving the partial derivatives are obtained in the explicit form using Eq. (A.5) as follows:

1. For $p_1 \equiv f$:

$$\begin{array}{rcl} \frac{\partial \mathbf{a}_{1,4}^t}{\partial f}\mathbf{X}_n & = & r_{11}(X_n - X_0) + r_{12}(Y_n - Y_0) + r_{13}(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial f}\mathbf{X}_n & = & r_{21}(X_n - X_0) + r_{22}(Y_n - Y_0) + r_{23}(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial f}\mathbf{X}_n & = & 0. \end{array} \tag{A.33}$$

2. For $p_2 \equiv x_0$:

$$\begin{array}{rcl} \frac{\partial \mathbf{a}_{1,4}^t}{\partial x_0}\mathbf{X}_n & = & r_{31}(X_n - X_0) + r_{32}(Y_n - Y_0) + r_{33}(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial x_0}\mathbf{X}_n & = & 0; \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial x_0}\mathbf{X}_n & = & 0. \end{array} \tag{A.34}$$

3. For $p_3 \equiv y_0$:

$$\begin{array}{rcl} \frac{\partial \mathbf{a}_{1,4}^t}{\partial x_0}\mathbf{X}_n & = & 0; \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial x_0}\mathbf{X}_n & = & r_{31}(X_n - X_0) + r_{32}(Y_n - Y_0) + r_{33}(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial x_0}\mathbf{X}_n & = & 0. \end{array} \tag{A.35}$$

4. For $p_4 \equiv X_0$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial X_0}\mathbf{X}_n &= -r_{11}f + r_{31}x_0; \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial X_0}\mathbf{X}_n &= -r_{21}f + r_{31}y_0; \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial X_0}\mathbf{X}_n &= -r_{31}. \end{aligned} \tag{A.36}$$

5. For $p_5 \equiv Y_0$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial Y_0}\mathbf{X}_n &= -r_{12}f + r_{32}x_0; \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial Y_0}\mathbf{X}_n &= -r_{22}f + r_{32}y_0; \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial Y_0}\mathbf{X}_n &= -r_{32}. \end{aligned} \tag{A.37}$$

6. For $p_6 \equiv Z_0$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial Z_0}\mathbf{X}_n &= -r_{13}f + r_{33}x_0; \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial Z_0}\mathbf{X}_n &= -r_{23}f + r_{33}y_0; \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial Z_0}\mathbf{X}_n &= -r_{33}. \end{aligned} \tag{A.38}$$

7. For $p_7 \equiv \omega$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial \omega}\mathbf{X}_n &= -a_3(Y_n - Y0) + a_2(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial \omega}\mathbf{X}_n &= -a_7(Y_n - Y0) + a_6(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial \omega}\mathbf{X}_n &= -a_{11}(Y_n - Y0) + a_{10}(Z_n - Z_0). \end{aligned} \tag{A.39}$$

8. For $p_8 \equiv \phi$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial \phi}\mathbf{X}_n &= (-a_9 f\cos\kappa + x_0\cos\phi)(X_n - X_0) \\ &+ (-a_{10} f\cos\kappa + x_0 a_9 \sin\omega)(Y_n - Y_0) \\ &+ (-a_{11} f\cos\kappa - x_0 a_9 \cos\omega)(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial \phi}\mathbf{X}_n &= (a_9 f\sin\kappa - y_0\cos\phi)(X_n - X_0) \\ &+ (a_{10} f\sin\kappa - y_0 a_9 \sin\omega)(Y_n - Y_0) \\ &+ (a_{11} f\sin\kappa + y_0 a_9 \cos\omega)(Z_n - Z_0); \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial \phi}\mathbf{X}_n &= -a_{11}(Y_n - Y_0) + a_{10}(Z_n - Z_0). \end{aligned} \tag{A.40}$$

9. For $p_9 \equiv \kappa$:

$$\begin{aligned} \frac{\partial \mathbf{a}_{1,4}^t}{\partial \kappa}\mathbf{X}_n &= (r_{21}(X_n - X_0) + r_{22}(Y_n - Y_0) + r_{23}(Z_n - Z_0))f; \\ \frac{\partial \mathbf{a}_{5,8}^t}{\partial \omega}\mathbf{X}_n &= -(r_{11}(X_n - X_0) + r_{12}(Y_n - Y_0) + r_{13}(Z_n - Z_0))f; \\ \frac{\partial \mathbf{a}_{9,12}^t}{\partial \omega}\mathbf{X}_n &= 0. \end{aligned} \tag{A.41}$$

Appendix B

More on a Fundamental Matrix

This appendix contains a slightly rewritten paper of Y. Li and G. Gimel'farb "On Estimation of Fundamental Matrix in Computational Stereo" that ap[peared in the *Proceedings of the Image and Vision Computing New Zealand (IVCNZ'98), Auckland, New Zealand, Nov. 1998*, pp.268–273. It addresses the problem of estimating a fundamental matrix from a given set of corresponding pixels in two perspective images of a 3D scene that form a stereopair. The 3×3 fundamental matrix of rank 2 determines the corresponding epipolar lines in both images. Experiments with real stereo pairs have established the main reasons for instabilities of the well-known linear estimation based on the 8-parameter representation of a matrix. Three alternative 8-parameter representations are compared and an enhanced estimation scheme that combines the linear estimation of the six parameters with the subsequent non-linear minimisation of the total distance between the corresponding pixels and relevant epipolar lines is proposed. This latter non-linear minimisation is done first by the direct exhaustion of the remaining two parameters and then by the local minimisation of the total distance in a vicinity of the minimum found by the exhaustion.

Two perspective images of a 3D scene are related by *epipolar geometry* described by a 3×3 singular *fundamental matrix* [3]–[17]. To obtain a weakly calibrated stereo pair [3], the matrix can be estimated from the known coordinates of corresponding pixels in two images.

Usually, the estimation is formulated as a constrained least-square problem of choosing the 3×3 matrix of rank 2 that yields the minimum total distance between the corresponding pixels and relevant epipolar lines in both images. The total distance depends non-linearly on the matrix components, so that a straightforward constrained minimisation, say, by gradient-based techniques, involves too cumbersome computations (especially, because the distance is multi-modal with respect to the desired matrix components). The problem can be simplified by replacing the distance by a particular quadratic form and using less complicated linear minimisation techniques based, for instance, on the eigenvector computations. But, the small value of the approximating quadratic form does not guarantee the small value of the actual distance, so that such approximate solutions are not robust with respect to random errors in the coordinates of corresponding pixels.

Here, we propose an enhanced linear estimation combining the traditional linear estimation of some matrix parameters with a subsequent non-linear minimisation of the total distance between the corresponding pixels and epipolar lines with respect to the remaining parameters. This enhanced technique is based on the well-known 8-parameter representation of the fundamental matrix of rank 2 [17]. The six parameters are obtained by the linear estimation, and the non-linear minimisation is done by the direct exhaustion of the two remaining parameters within a sufficiently large range of their values and by the local minimisation in a vicinity of the minimum distance found by the exhaustion. Experiments with natural stereopairs show that the proposed approach yields a robust estimate of a fundamental matrix.

This appendix is organised as follows. Section B.1 overviews basic features of a fundamental matrix, describes the known linear estimation scheme, compares the three alternative 8-parameter represenations of a fundamental matrix, and proposes the enhanced linear estimation of a matrix. Section **??** presents some experimental results that reveal the main reasons for the non-robustness of the traditional linear estimation techniques to be ovecome by the above enhanced linear estimation.

B.1 Estimation of a fundamental matrix

Basic features of fundamental matrix. Epipolar constraint in stereo means that, for each pixel m_{k1} in the left image of a stereopair, the corresponding pixel m_{k2} in the right image lies on the relevant epipolar line which is uniquely specified by the pixel m_{k1}. Alternatively, for each pixel m_{k2} in the right image, the corresponding pixel m_{k1} lies on the relevant epipolar line specified by the pixel m_{k2}.

This constraint is quantitatively expressed by the fundamental 3×3 matrix of rank 2:

$$\mathbf{F} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_4 \\ a_7 & a_8 & a_9 \end{pmatrix},$$

that links the homogeneous coordinates of the corresponding pixels as follows [4]:

$$\mathbf{m}_{k2}^T \mathbf{F} \mathbf{m}_{k1} = 0. \tag{B.1}$$

Here, the vectors $\mathbf{m}_{kj}^T = [x_{kj}, y_{kj}, 1] : \ j = 1, 2$, represent homogeneous coordinates of k-th pair of the corresponding pixels in a stereopair, $j = 1, 2$ are the left and the right image, respectively, and T denotes the transposition. The vectors

$$\begin{aligned} \mathbf{l}_{k1} &\equiv [\mathbf{m}_{k2}^T \mathbf{F}]^T \\ &= \begin{bmatrix} l_{k11} = a_1 x_{k2} + a_4 y_{k2} + a_7 \\ l_{k12} = a_2 x_{k2} + a_5 y_{k2} + a_8 \\ l_{k13} = a_3 x_{k2} + a_6 y_{k2} + a_9 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{l}_{k2} &\equiv \mathbf{F}\mathbf{m}_{k1} \\ &= \begin{bmatrix} l_{k21} = a_1 x_{k1} + a_2 y_{k1} + a_3 \\ l_{k22} = a_4 x_{k1} + a_5 y_{k1} + a_6 \\ l_{k23} = a_7 x_{k1} + a_8 y_{k1} + a_9 \end{bmatrix} \end{aligned}$$

represent the coefficients of the corresponding epipolar lines that should pass through the pixels m_{k1} and m_{k2}, respectively.

That the fundamental matrix $\mathbf{F}$ is of rank 2 is explicitly demonstrated by involving into Eq. (B.1) the coordinates of the epipoles $\mathbf{e}_1 = [x_{e1}, y_{e1}, 1]^T$ and $\mathbf{e}_2 = [x_{e2}, y_{e2}, 1]^T$ in both the images [17]:

$$\mathbf{F} = \begin{pmatrix} a_1 & a_2 & -x_{e1}a_1 - y_{e1}a_2 \\ a_4 & a_5 & -x_{e1}a_4 - y_{e1}a_5 \\ -x_{e2}a_1 - y_{e2}a_4 & -x_{e2}a_2 - y_{e2}a_5 & \begin{array}{l} x_{e1}x_{e2}a_1 + y_{e1}x_{e2}a_2 \\ +x_{e1}y_{e2}a_4 + y_{e1}y_{e2}a_5 \end{array} \end{pmatrix}. \tag{B.2}$$

Linear estimation: a known technique. An error of the epipolar geometry in the image j is given by the distance from a pixel m_{kj} in this image to a relevant epipolar line $\mathbf{l}_{kj}^T = (l_{kj1}, l_{kj2}, l_{kj3})$ as follows:

$$d_{kj} = d_{kj}\left(\mathbf{m}_{kj}, \mathbf{l}_{kj}\right) = \frac{\mathbf{m}_{k2}^T \mathbf{F} \mathbf{m}_{k1}}{\left(l_{kj1}^2 + l_{kj2}^2\right)^{\frac{1}{2}}}. \tag{B.3}$$

This distance is invariant to uniform scaling of the matrix components $a_1, \ldots, a_9$. Because the denominator in Eq.(B.3) is independent of a_9, the fundamental matrix $\mathbf{F}$ can be normalised so that the following constraint holds: $\sum_{i=1}^{8} a_i^2 = 1$.

If the corresponding pixels $k = 1, \ldots, n$ are known, the matrix $\mathbf{F}$ can be estimated by minimising the total square error in both the images:

$$E = \sum_{k=1}^{n} \left(d_{k1}^2 + d_{k2}^2\right) \tag{B.4}$$

with respect to the components $a_1, \ldots, a_9$ under the above constraint of rank 2.

To avoid the computationally complex non-linear minimisation of the total error E, various linear approximations have been proposed in [5, 16, 17]. One of most popular linear schemes is based on minimising the unnormalised total error:

$$E_{\text{appr}} = \sum_{k=1}^{n} \left(\mathbf{m}_{k2}^T \mathbf{F} \mathbf{m}_{k1}\right)^2 \tag{B.5}$$

under the asymmetric 8-parameter representation of the matrix of rank 2 proposed in [5]:

$$\mathbf{F} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ \alpha a_1 + \beta a_4 & \alpha a_2 + \beta a_5 & \alpha a_3 + \beta a_6 \end{pmatrix}. \tag{B.6}$$

Eq. (B.2) suggests that the factors $\alpha = -x_{e2}, \beta = -y_{e2}$, respectively. The minimisation is carried out by the 2-stage block relaxation as follows. (*i*) For given values (α, β), the six parameters $\mathbf{a}^T = [a_1, \ldots, a_6]$ can be estimated as follows:

$$\min_{\mathbf{a}} E_{\text{appr}}(\alpha, \beta) = \mathbf{a}^T \Gamma_{\alpha,\beta} \mathbf{a} \quad \text{under} \quad \mathbf{a}^T \mathbf{a} = 1 \tag{B.7}$$

where the symmetric matrix $\Gamma_{\alpha,\beta}$ is computed from the known coordinates of the corresponding pixels and the parameters α, β. In such a case, the desired vector $\mathbf{a}$ is obtained as the eigenvector of the matrix $\Gamma_{\alpha,\beta}$ with the smallest eigenvalue. (*ii*) Under the known vector $\mathbf{a}$, the total error of Eq. (B.5) is reduced to a quadratic form with respect to α and β, and these latter can be found by solving a system of two linear equations obtained by differentiating Eq. (B.5) with respect to α and β.

This approach is computationally simple but has several drawbacks.

- Usually, the matrix $\Gamma_{\alpha,\beta}$ is ill-defined, so the initial coordinates should be normalised in a specific way to obtain computationally stable results.

- The minimum unnormalised error of Eq. (B.5) may not correspond to the desired minimum of the total error in Eq. (B.4). To avoid this drawback, several iterative schemes based on the total weighted square error where the weights are specified by the denominators of Eq. (B.3) have been proposed [17]. But, such an iterative weighting stops usually in one of the local minima of the weighted error.

- The matrix representation of Eq. (B.6) results in singular parameter values for the ideal horizontal or vertical stereopairs. For example, the ideal horizontal pair has the following fundamental matrix:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.707 \\ 0 & -0.707 & 0 \end{pmatrix}$$

  so that the above 8 parameters are as follows: $\alpha = -\infty$, $\beta = 0$, $a_1 = a_2 = a_3 = a_4 = a_5 = 0$, $a_6 = 0.707$, but simultaneously $\alpha a_1 = 0$, $\alpha a_2 = -0.707$, and $\alpha a_3 = 0$.

Enhanced linear estimation. To avoid these drawbacks, we combine the computationally simple linear and non-linear techniques as follows. Instead of iterative block relaxation by the parameters $a_1, \ldots, a_6$ and α, β, we simply use the direct exhaustion of α and β values in a broad range which is sufficient to find a vicinity of the global minimum of the total error of Eq.(B.4).

To exclude possible singularities, we use the following two 8-parameter representations of $\mathbf{F}$ simultaneously with Eq. B.6. The representation

$$\mathbf{F} = \begin{pmatrix} a_1 & a_2 & a_3 \\ \alpha a_1 + \beta a_7 & \alpha a_2 + \beta a_8 & \alpha a_3 + \beta a_9 \\ a_7 & a_8 & a_9 \end{pmatrix}, \tag{B.8}$$

is suitable for the ideal vertical stereopairs (here, $\alpha = -\frac{x_{e2}}{y_{e2}}$ and $\beta = -\frac{1}{y_{e2}}$ as follows from Eq. (B.2)), and the representation

$$\mathbf{F} = \begin{pmatrix} \alpha a_4 + \beta a_7 & \alpha a_5 + \beta a_8 & \alpha a_6 + \beta a_9 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}, \tag{B.9}$$

is convenient for the ideal horizontal stereopairs (here, $\alpha = -\frac{y_{e2}}{x_{e2}}$ and $\beta = -\frac{1}{x_{e2}}$). Singularities in these latter representations exist when $y_{e2} = 0$ in Eq.(B.8) or $x_{e2} = 0$ in Eq.(B.9) but these values are not typical for usual vertical or horizontal stereopairs, respectively.

After computing the eight parameters in Eqs. B.6, B.8, and B.9, we apply the same renormalisation $\sum_{i=1}^{8} a_i^2 = 1$ to all the resulting matrices.

B.2 Experimental results and conclusions

Experiments with the natural stereopairs (one of them, taken from [18], with 128 known corresponding pixels is used in Figures B.1 – B.3) show that the total error of Eq.(B.4) has multiple local minima with respect to the parameters α and β if the other six parameters $\mathbf{a} = [a_1, \ldots, a_6]$ are found by solving Eq. (B.7). That is why a vicinity of the desired global minimum of this error has to be searched by the direct exhaustion, and only then the local minimisation techniques such as the above-mentioned block relaxation in $\mathbf{a}$ and (α, β) can be used for refining these values.

The enhanced linear estimation results in the slightly different estimates for the matrix representations of Eqs. (B.6), B.8), and (B.9) presented in Table B.1. Because of a finite range of exhausting the parameters α, β the minimum total error for both the representations in Eq.(B.8) and Eq.(B.9) has been found at the borders of the range. But, in this case this approach suggests how to extend the search range as to approach the globally minimum error.

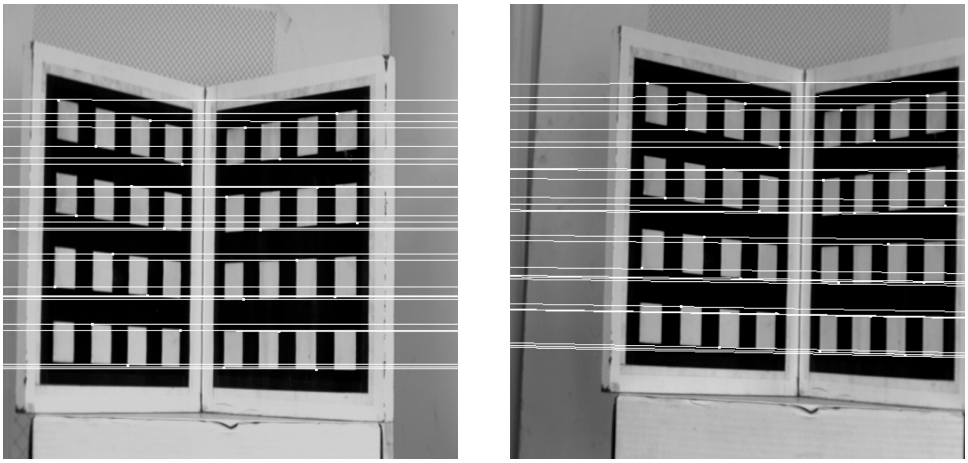


Figure B.1: Epipolar lines for the matrix represented by Eq. B.6 (the mean square distance error: 0.01 pixel; the maximum square distance error: 0.122 pixel).

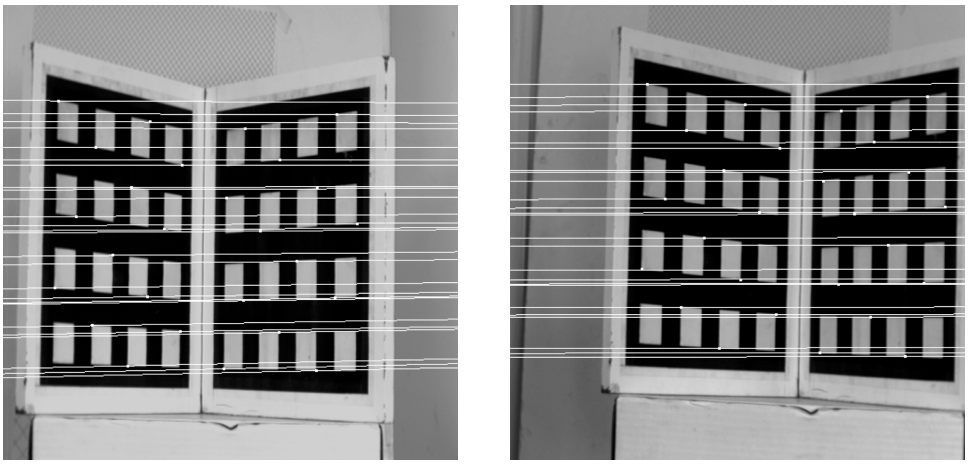


Figure B.2: Epipolar lines for the matrix represented by Eq. B.8 (the mean square distance error: 0.02 pixel, the maximum square distance error: 0.145 pixel).

Table B.1: Estimated fundamental matrices.

$\alpha = 100, \beta = 1$ in Eq.(B.6)			$\alpha = 500, \beta = 0$ in Eq.(B.8)			$\alpha = 0, \beta = 0$ in Eq.(B.9)		
0	0	0.012	0	0	−0.001	0	0	0
0	0	0.716	0	0	−0.712	0	0	−0.712
−0.001	−0.698	10.5	−0.01	0.702	−11.5	−0.01	0.702	−11.6

Stereo pairs with the overlaid epipolar lines corresponding to Table B.1 are shown in Figures B.1-B.3. Here, all the three cases give practically the same precision of the resulting epipolar lines. Tables B.2 and B.3 present the mean values of the matrix components $a_1, \ldots, a_9$ and their standard deviations with respect to the ideal ones. These results, obtained under random variations of the coordinates of the corresponding pixels within the two ranges: $[-1, 1]$ and $[-5, 5]$ pixels, confirm the robustness of the proposed approach.

Table B.2: Mean values (μ) and standard deviations (σ) of the matrix components under random variations of the coordinates of the corresponding pixels in the range $[-1, 1]$ pixels.

		$10^7 a_1$	$10^7 a_2$	$10^5 a_3$	$10^5 a_4$	$10^5 a_5$	$10^3 a_6$	$10^3 a_7$	$10^3 a_8$	a_9
(B.6)	μ	1.5	−676	1090	−0.08	0.2	716	1.7	−698	10.6
	σ	1.3	1.4	101	0.7	1	2.6	2.1	2.7	0.736
(B.8)	μ	1.3	−0.05	−142	6.6	−0.3	−712	−9.6	702	−11.5
	σ	0.1	0.09	0.3	0.7	0.4	1.3	1.8	1.4	0.547
(B.9)	μ	−0.02	0.7	1	6.6	−0.2	−712	−11	702	−11.6
	σ	0.03	3	8.4	0.7	1	2.6	1.8	2.6	0.725

The above experiments show that the described approach allows to obtain robust estimates of a fundamental matrix in a computationally simple way that can be easily implemented in practice. All the three representations give fairly similar results but the representation in Eq. (B.8) seems to be slightly more stable under the random noise. The representation in Eq.(B.9) is most convenient when a given stereopair is almost horizontal.

Table B.3: Mean values (μ) and standard deviations (σ) of the matrix components under random variations of the coordinates of the corresponding pixels in the range $[-5, 5]$ pixels.

		$10^7 a_1$	$10^7 a_2$	$10^5 a_3$	$10^5 a_4$	$10^5 a_5$	$10^3 a_6$	$10^3 a_7$	$10^3 a_8$	a_9
(B.6)	μ	1.3	-675	1090	-0.01	0.2	716	1.4	-698	10.4
	σ	4	6	104	3	4	10	7.2	10.3	2.79
(B.8)	μ	1.3	-0.04	-142	6.5	-0.2	-712	-9.4	702	-11.7
	σ	0.5	0.5	1	3	2	6.7	7.2	6.9	2.29
(B.9)	μ	-0.01	2.5	-3	6.6	-0.2	-712	-10.7	702	-11.6
	σ	0.2	22	59	2.7	3.7	9.9	7.1	10	2.76

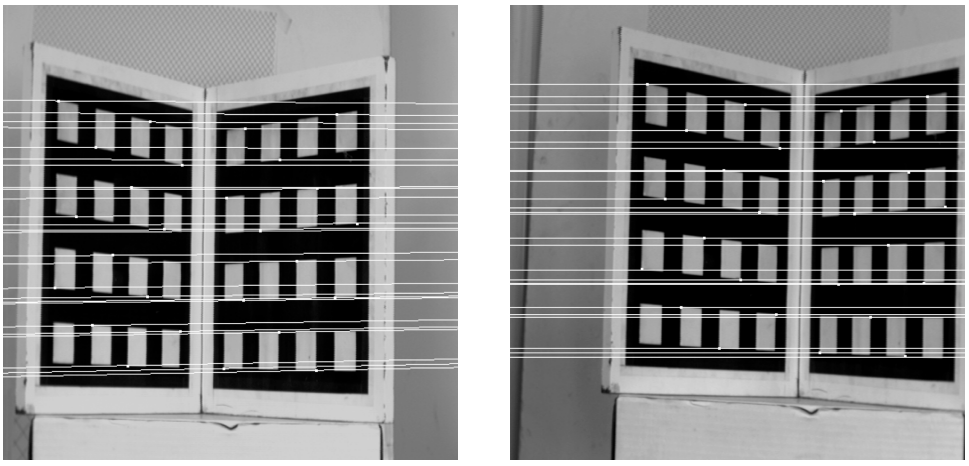


Figure B.3: Epipolar lines for the matrix of Eq. B.9 (the mean square distance error: 0.02 pixel, the maximum square distance error: 0.138 pixel).

Bibliography

[1] C. M. Brown and D. Terzopoulos, *Real-Time Computer Vision*. Cambridge University Press: Cambridge (1994).

[2] E. R. Dougherty and P. A. Laplante, *Introduction to Real-Time Imaging*. SPIE: Bellingham (1995).

[3] O. D. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini (ed.), *Proc. 2nd European Conf. on Computer Vision*, Santa Margerita Ligure, Italy, May 19-22, 1992. *Lecture Notes in Computer Science* **588**. Springer-Verlag, Berlin (1992) 563–578.

[4] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press: Cambridge (1993).

[5] O. D. Faugeras and Q.-T. Luong, The fundamental matrix: theory, algorithms, and stability analysis. *Int. Journal of Computer Vision* **17** (1996) 43–75.

[6] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, Camera self-calibration: theory and experiments. In G. Sandini (ed.), *Proc. 2nd European Conf. on Computer Vision*, Santa Margerita Ligure, Italy, May 19-22, 1992. *Lecture Notes in Computer Science* **588**. Springer-Verlag, Berlin (1992) 321–334.

[7] R. Haralick & L. Shapiro, "Image Segmentation Techniques". *Comput. Vision, Graphics, and Image Processing*, vol. 29 (1985).

[8] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Addison-Wesley: Reading, Mass. Vol. 1 (1992), Vol. 2 (1993).

[9] R. Klette, K. Schlüns, and A. Koschan, *Computer Vision: Three-Dimensional Data from Images*. Springer: Singapore (1998).

[10] R. Kjeldsen and J. Kender, "Finding Skin in Color Images". 2nd *Int. Conf. on Automatic Face and Gesture Recognition, Oct. 13–16, 1996, Killington, Vermont, USA*, IEEE Computer Society: Electronic Edition (1996) 312–317.

[11] M. Lievin and F. Luthon, "A Hierarchical Segmentation Algorithm for Face Analysis. Application to Lipreading". *IEEE Conference on Multimedia and Expo, New York City, New York, July 2000. Proc*. TP8.04 (2000) 1085–1088.

[12] S. I. Olsen, Epipolar line estimation. In G. Sandini (ed.), *Proc. 2nd European Conf. on Computer Vision*, Santa Margerita Ligure, Italy, May 19-22, 1992. *Lecture Notes in Computer Science* **588**. Springer-Verlag, Berlin (1992) 307–311.

[13] R. Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation* **RA-3**(4): 323-344 (1987).

[14] H. Wechsler, *Computer Vision*. Academic Press: Boston (1990).

[15] A. Wu, M. Shah, and N. da Vitoria Lobo, "A Virtual 3D Backboard: 3D Finger Tracking using a Single Camera". *4*th *IEEE Int. Conf. on Automatic Face and Gesture Recognition, March 26–30, 2000, Grenoble, France. Proc.*. IEEE Computer Society Press (2000) 535–543.

[16] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong, *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*. INRIA Sophia Antipolis, Research Report 2273 (May 1994).

[17] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review. *Int. Journal of Computer Vision* **27** (1998) 161–195.

[18] Z. Zhang, http://www.inria.fr/robotvis/personnel/zzhang/zzhang-eng.html. Demo for stereo calibration.