

Active Contours

Andrew Blake and Michael Isard



Springer

Active Contours

Springer

London

Berlin

Heidelberg

New York

Barcelona

Budapest

Hong Kong

Milan

Paris

Santa Clara

Singapore

Tokyo

Andrew Blake and Michael Isard

Active Contours

**The Application of Techniques from
Graphics, Vision, Control Theory and Statistics
to Visual Tracking of Shapes in Motion**



Springer

Andrew Blake and Michael Isard
Department of Engineering Science
University of Oxford
Parks Road
Oxford OX1 3PJ

Cover image created by Andrew Blake with the assistance of Josephine Sullivan

ISBN-13:978-1-4471-1557-1 e-ISBN-13:978-1-4471-1555-7
DOI: 10.1007/978-1-4471-1555-7

British Library Cataloguing in Publication Data
Blake, Andrew

Active contours : the application of techniques from
graphics, vision, control theory and statistics to visual
tracking of shapes in motion

1. Computer vision 2. Pattern recognition systems

I. Title II. Isard, Michael

006.4

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

© Springer-Verlag London Limited 1998

Softcover reprint of the hardcover 1st edition 1998

The use of registered names, trademarks etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed and bound at Cambridge University Press, Cambridge, England
34/3830-543210 Printed on acid-free paper

Contents

1	Introduction	1
1.1	Organisation of the book	3
1.2	Applications	5
2	Active shape models	25
2.1	Snakes	26
2.2	Deformable templates	32
2.3	Dynamic contours	34
I	Geometrical Fundamentals	39
3	Spline curves	41
3.1	B-spline functions	42
3.2	Finite bases	45
3.3	Multiple knots	46
3.4	Norm and inner product for spline functions	47
3.5	B-spline parametric curves	53
3.6	Curves with vertices	54
3.7	Control vector	57
3.8	Norm for curves	58
3.9	Areas and moments	63
4	Shape-space models	69
4.1	Representing transformations in shape-space	70
4.2	The space of Euclidean similarities	75

4.3	Planar affine shape-space	76
4.4	Norms and moments in a shape-space	79
4.5	Perspective and weak perspective	81
4.6	Three-dimensional affine shape-space	88
4.7	Key-frames	90
4.8	Articulated motion	93
5	Image processing techniques for feature location	97
5.1	Linear scanning	99
5.2	Image filtering	100
5.3	Using colour	104
5.4	Correlation matching	105
5.5	Background subtraction	110
6	Fitting spline templates	115
6.1	Regularised matching	115
6.2	Normal displacement in curve fitting	120
6.3	Recursive solution of curve-fitting problems	126
6.4	Examples	134
7	Pose recovery	141
7.1	Calculating the pose of a planar object	141
7.2	Pose recovery for three-dimensional objects	149
7.3	Separation of rigid and non-rigid motion	153
II	Probabilistic Modelling	157
8	Probabilistic models of shape	159
8.1	Probability distributions over curves	160
8.2	Posterior distribution	164
8.3	Probabilistic modelling of image features	169
8.4	Validation gate	172
8.5	Learning the prior	174
8.6	Principal Components Analysis (PCA)	176

9	Dynamical models	185
9.1	Some simple dynamical prior distributions	187
9.2	First-order Auto-regressive processes	193
9.3	Limitations of first-order dynamical models	196
9.4	Second-order dynamical models	200
9.5	Second-order AR processes in shape-space	203
9.6	Setting dynamical parameters	205
10	Dynamic contour tracking	213
10.1	Temporal fusion by Kalman filter	213
10.2	Tracking performance	220
10.3	Choosing dynamical parameters	225
10.4	Case study	231
11	Learning motion	235
11.1	Learning one-dimensional dynamics	236
11.2	Learning AR process dynamics in shape-space	242
11.3	Dynamical modes	247
11.4	Performance of trained trackers	250
12	Non-Gaussian models and random sampling algorithms	255
12.1	Factored sampling	257
12.2	The CONDENSATION algorithm	259
12.3	An observation model	262
12.4	Applications of the CONDENSATION algorithm	267
	Appendix	281
A	Mathematical background	281
A.1	Vectors and matrices	281
A.2	B-spline basis functions	284
A.3	Probability	294
B	Stochastic dynamical systems	297
B.1	Continuous-time first-order dynamics	297
B.2	Second-order dynamics in continuous time	299

B.3 Accuracy of learning	300
C Further shape-space models	303
C.1 Recursive synthesis of shape-spaces	303
Glossary of notation	311
Bibliography	315
Author Index	335
Index	341

Foreword

The field of computer vision has its sights set on nothing less than enabling computers to see. This monumental challenge has absorbed many creative minds over the course of more than three decades. A basic premise held by the computer vision community is that vision may be understood in precise computational terms, and that doing so raises the possibility of engineering camera-equipped computer systems with human-like perceptual abilities. Once envisioned only in science fiction, powerful machine vision systems are now more than ever poised to become science fact. This is due in part to the advent of increasingly potent microprocessors, as predicted by Moore's law, and in part to the slow but steady unraveling, on multiple scientific fronts, of the mystery that is visual perception in living systems.

Model-based vision is a major trend in the field that approaches computational problems attendant to vision using mathematical models. To see familiar objects as normal people evidently do with ease, computer vision systems must be able to analyze object shape and motion in real time. To this end, in the early 1980s, my colleagues and I introduced a family of mathematical models, known as "deformable models". The motivation was to formulate visual models that unify the representation of shape and motion by combining geometry and physics; in particular, free-form (spline) geometry and the dynamics of elastic curves, surfaces, and solids. We anticipated that deformable models would lead to vision systems capable of interpreting video sequences in terms of rigid and nonrigid objects moving before the camera. Perhaps the simplest deformable model, deformable contours confined to the plane, also known as "active contours" or "snakes", quickly gained popularity following early public demonstrations of these contours actively conforming to the shapes and tracking the motions of object boundaries in video sequences.

I have admired Andrew Blake and his work for many years. His contribution to computer vision is undeniable. A very readable author, Blake's book on *Visual Reconstruction* has become a classic in the field. It gives me great pleasure to see the concept of active contours developed to the remarkable degree evident in this, his latest book, which is authored with his talented student, Michael Isard.

In a characteristically no-nonsense, mathematically solid treatment, Blake and Isard take the subject of active contours to new heights of theoretical sophistication and practical application. The latter addresses the difficult task of visually tracking the motions of a variety of complex objects captured by a video camera feeding a frame-rate video digitizer. The impressive technical achievement of the book is a

novel, probabilistic interpretation of active contours built on a geometric substrate that combines B-spline curve bases with shape spaces defined by global deformations. This combination leads to a new class of very fast and highly robust (non-Gaussian) active contour propagation algorithms. Another noteworthy achievement is the ability of these new tracking algorithms to learn the complex motions of specific objects through observation, thereby automatically tuning the tracker with greater selectivity to objects of interest, further enhancing its robustness.

This book defines the state-of-the-art of contour-based object tracking algorithms. It is required reading for anyone interested in computer vision theory and in the design of working computer vision systems.

Demetri Terzopoulos
November, 1997

Preface

In the seventies and eighties, interest in Computer Vision was concentrated on the development of general purpose seeing machines. There was wide agreement on research priorities, developing “bottom-up” computer algorithms that would organise the raw intensity values in images into a more compact form. The purpose of this was not just to compress the data but also to extract its salient features. Salient features could include corners, edges and surface fragments, to be used in identifying objects and deducing their positions. However, experience suggests strongly that general purpose vision is too difficult a goal for the time being.

If general purpose vision is abandoned, what alternative approach could be taken? One answer is that generality can be abated by introducing some “prior” knowledge — knowledge that is specific to the objects that the computer is expected to see. An extreme form of this approach is exemplified by automatic visual inspection machines of the kind used on factory assembly lines. In that context, it is known in advance precisely what objects are to be inspected — it is rare, after all, for potatoes streaming along a conveyor to give way, without notice, to a crop of spanners or chocolate bars. When computer hardware and software are specialised entirely to deal with one object, phenomenal performance can be obtained. A striking example is the “Niagara” machine (Sortex, UK Ltd) for sorting rice grains which “sees” 70,000 grains every second and almost literally spits out the rejects.

It is a commonly held view that it is hard to make progress in research by building such specialised machines because general principles are lost to engineering detail. That is a fair point but by no means, in our view, outlaws the use of prior knowledge about shape in computer vision research. Instead, we would argue, scientific principles for representing prior knowledge need to be developed. Then, when a new problem area is addressed, the principles can be applied to “compile” a new vision system as rapidly as possible. This includes such issues as how to represent classes of shapes that are defined loosely. Potatoes, for instance, might be characterised as roundish but with substantial size variations, with or without knobs. On the other hand, the class of human faces could be represented in terms of a common basic layout, but with considerable variation in the sizes and separations of features. Modelling classes of shapes, their variability and their motion is one of the principal themes of the book. The use of those models to help interpret moving images is the other central theme.

We have tried to present ideas about shape and motion in a way that will be readable not only by specialists, but also by those who are not regularly immersed in

the ideas of machine vision. In particular we would hope that those with backgrounds in graphics or signal processing or neural computing would find the book a useful and accessible guide.

Acknowledgements

We have enjoyed and are grateful for discussions at various times and for proof-reading by Benedicte Bascle, Fiona Blake, Mike Brady, Roger Brockett, Roberto Cipolla, Rupert Curwen, Ron Daniel, Colin Davidson, Hugh Durrant-Whyte, Andrew Fitzgibbon, Robert Kaucic, Pete Lindsey, John Kent, John MacCormick, Elisa Martínez Marroquín, David Mumford, David Murray, Alison Noble, Ben North, Ian Reid, David Reynard, Brian Ripley, Jens Rittscher, Simon Rowe, Steve Smith, Lionel Tarassenko, Andrew Wildenberg, Alan Yuille and Andrew Zisserman. We would like to acknowledge support at various stages of the work from the EPSRC and the EU, and experimental assistance from Sarah Blake.

Part I

Geometrical Fundamentals

Chapter 7

Pose recovery

In certain three-dimensional applications (chapter 1), such as the 3D mouse in figure 1.16 on page 20, a shape-vector $\mathbf{X}$ is used to compute pose, in that case the position and attitude of the hand. Similarly, in facial animation, it is desirable to compute the attitude of the head, independently of expression if possible. The problem is to convert a shape-vector $\mathbf{X}$, from a planar or three-dimensional shape-space respectively, into three-dimensional translation $\mathbf{R}_c$ and rotation R .

7.1 Calculating the pose of a planar object

In the planar affine space, pose is recovered from a shape-vector $\mathbf{X}$ by first obtaining affine parameters $\mathbf{u}, M$ via (4.5) on page 78. Then $\mathbf{u}, M$ are used to obtain pose parameters. This is fairly straightforward under orthographic projection. For large fields of view however it is necessary to use weak perspective and then the pose-recovery computation must be elaborated to compensate for the obliqueness of projection in the periphery of the field of view.

Orthographic projection

First, a solution is presented for the pose of an object under orthographic projection, suitable for use when the field of view is significantly smaller than 1 radian. Given a planar affine vector $\mathbf{X}$, the algorithm computes object pose as the position vector $\mathbf{R}_c = (X_c, Y_c, Z_c)^T$ of the object's centre and the orientation as a rotation R relative to the reference pose. First the algorithm is given, then a short proof (optional).

1. Compute $\mathbf{u}$ and M from

$$\mathbf{X} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T.$$

2. Compute the eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ and eigenvalues λ_1, λ_2 (both must be positive and are ordered so that $\lambda_1 \geq \lambda_2$) of the matrix MM^T .

- 3.

$$Z_c = f / \sqrt{\lambda_1}$$

- 4.

$$\cos \theta = \left(\frac{Z_c}{f} \right)^2 \det M$$

- 5.

$$(\cos \phi, \sin \phi)^T = \mathbf{v}_1$$

- 6.

$$R(\psi) = S_y(1/\cos \theta) R(-\phi) M \frac{Z_c}{f}$$

$$\text{where } R(\psi) \equiv \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \text{ and } S_y(\mu) \equiv \begin{pmatrix} 1 & 0 \\ 0 & \mu \end{pmatrix}.$$

- 7.

$$\text{Combined rotation: } R = R_z(\phi) R_x(\theta) R_z(\psi).$$

- 8.

$$\text{Translation: } \mathbf{R}_c = (X_c, Y_c, Z_c)^T \text{ where } X_c = u_1 Z_c / f, \quad Y_c = u_2 Z_c / f.$$

Figure 7.1: Algorithm for recovery of the pose of a planar object.

The algorithm is summarised in figure 7.1 and a commentary follows. Given $\mathbf{u}, M$ from step 1 of the algorithm, we first work on M to recover the object's attitude as represented by the rotation matrix R . It is well known that a three-dimensional rota-

tion can be decomposed in terms of three *Euler angles* θ, ϕ, ψ appearing in a sequence of three standard rotations. It is particularly convenient here to use a decomposition that is aligned with the image plane:

$$R(\phi, \theta, \psi) = R_z(\phi)R_x(\theta)R_z(\psi). \quad (7.1)$$

This should be read from right to left as a z -rotation through an angle ψ followed by a rotation through angle θ about the x -axis and finally a z -rotation through angle ϕ , as illustrated in figure 7.2. Our algorithm recovers θ, ϕ and ψ so that the rotation matrix can be calculated as in (7.1). First Z_c, ϕ, θ and ψ are computed in steps 3–6. Note that a redundancy in ϕ arises from the indeterminacy of the sign of eigenvalue $\mathbf{v}_1$: for a given rotation R , ϕ can equally well be replaced by $\phi + 180^\circ$ (the substitution $\phi \rightarrow 180^\circ + \phi, \theta \rightarrow -\theta, \psi \rightarrow 180^\circ + \psi$ leaves R unchanged). Finally, the focal length f of the camera is known, so the image-plane components of translation $\mathbf{R}_c$ can be recovered directly in step 8. The result of applying this algorithm to a sequence of images of a moving hand are shown in figure 7.3.

Ambiguities

There are two kinds of ambiguity in the recovered parameter. The first is a straightforward indeterminacy in linear scale. If (X_c, Y_c, Z_c) was doubled and accompanied by a doubling in size of the object, there would be no visible effect on the image contour. The scale factor is fixed here by our convention in orthographic perspective that $Z_c = f$ in the standard view.

The second and more complex ambiguity is apparent in step (4) above: since $\cos \theta = \cos(-\theta)$, there must be two possible solutions $\pm\theta$. Unlike the indeterminacy of ϕ which represents simply an alternative representation of a given physical transformation, this is a genuine ambiguity, a reversal as illustrated in figure 7.4. In the 3D mouse application, for example, this could result in a “flip” of the object controlled by the mouse. One practical solution is to arrange for the camera to be oblique to the table so that the hand avoids the “singular” orientation $\theta = 0$, when image and object planes are parallel. Then at each time-step, the value $\pm\theta$ is chosen which give θ, ϕ and ψ closest to their values at the previous time-step.

Derivation of pose-recovery algorithm. The key point is that the effect of a rotation $R_x(\theta)$, on the image of a planar object lying on an xy plane, is to compress the image along



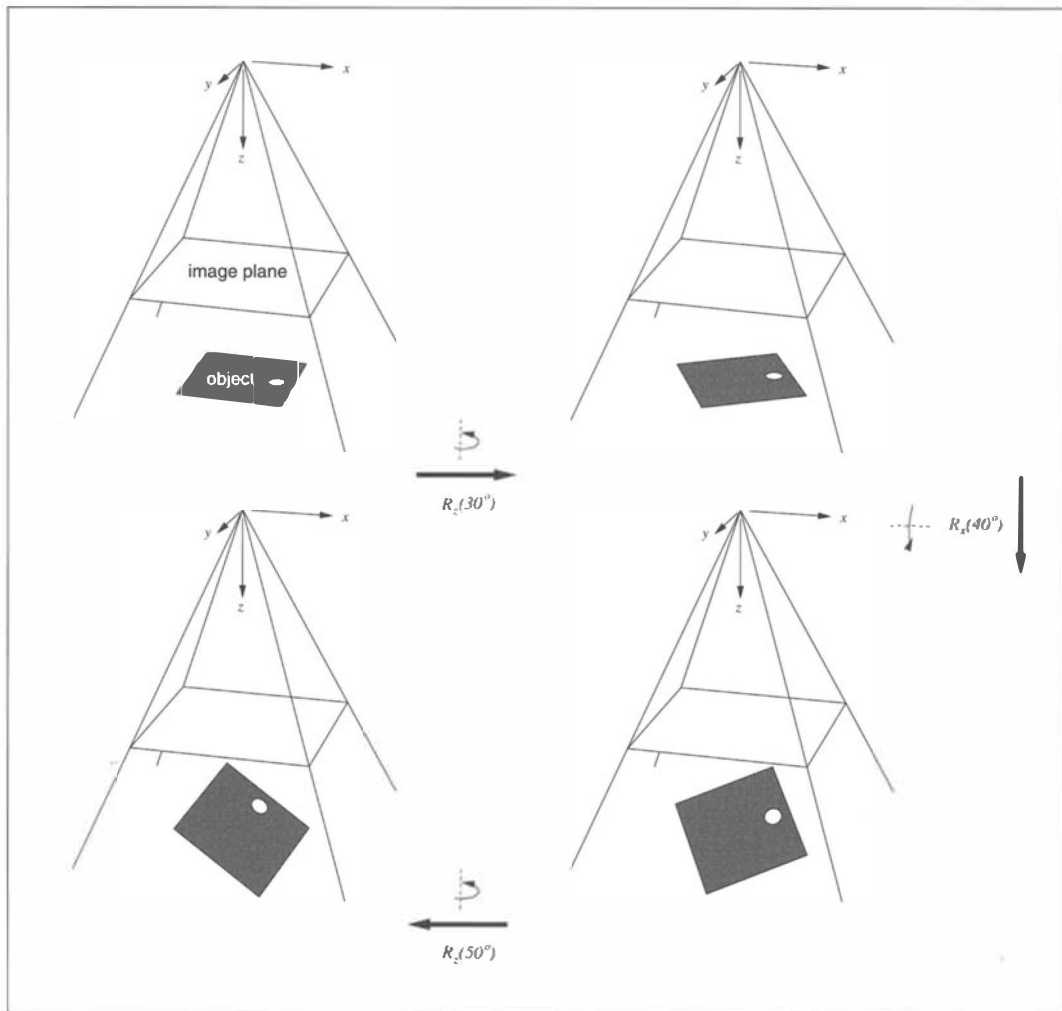


Figure 7.2: Euler angles. The figure shows Euler angles to describe rotations in the camera coordinate frame. A general rotation R is expressed as a sequence (7.1) and the case $\phi = 50^\circ$, $\theta = 40^\circ$, $\psi = 30^\circ$ is illustrated.

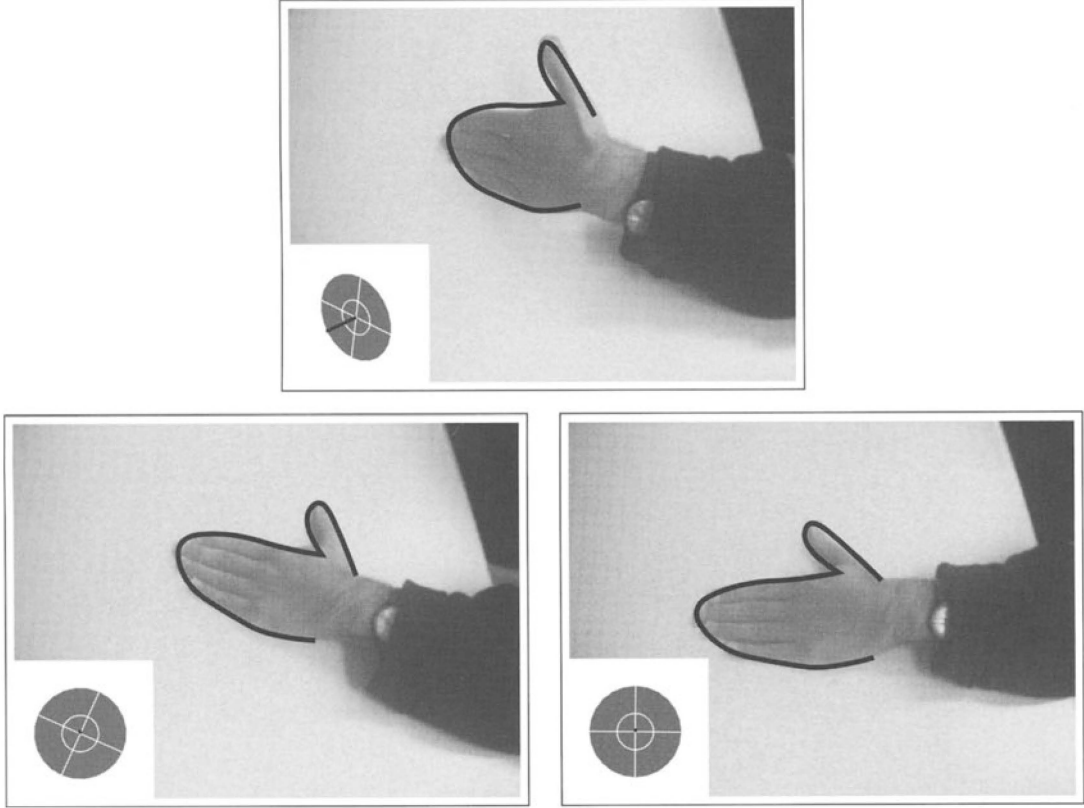


Figure 7.3: Computing pose. *The outline of a hand is tracked in a planar affine shape-space and the pose-recovery algorithm described above is applied. Computed pose is displayed using the “target” icon in the corner of each image.*

the y -axis, that is to apply a scaling transformation $S_y(\cos \theta)$. In that case, the affine image transformation due to the Euler angle transformations (7.1) and to distance scaling is:

$$M = \frac{f}{Z_c} R_z(\phi) S_y(\cos \theta) R_z(\psi).$$

The algorithm given above is simply a decomposition of this sequence, using the fact that

$$MM^T = \left(\frac{f}{Z_c} \right)^2 R_z(\phi) S_y(\cos^2 \theta) R_z(-\phi)$$

which is in diagonal form.



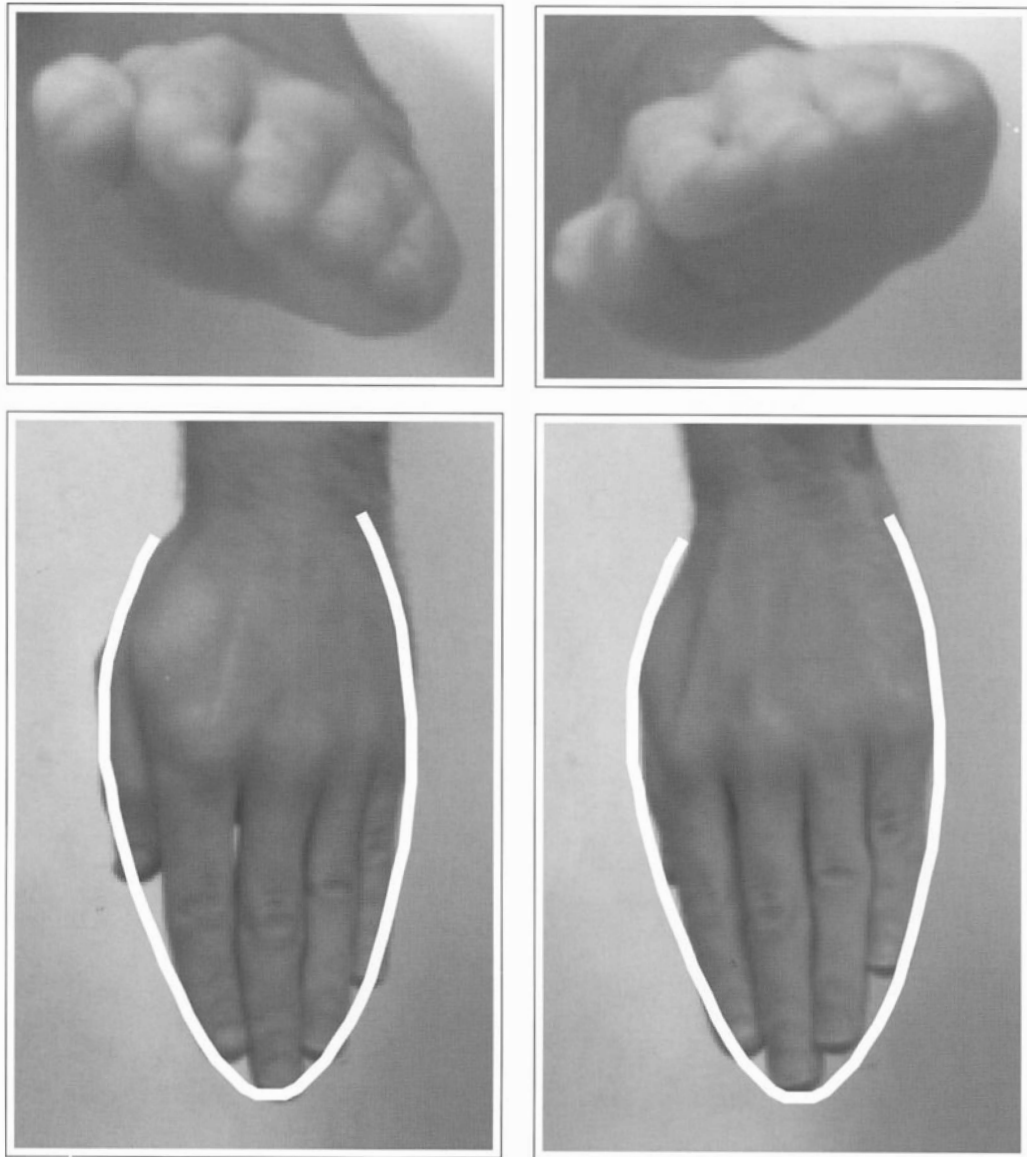


Figure 7.4: Ambiguity of pose recovery. *A hand in two alternative positions with equal and opposite slant (top row) gives rise to almost identical outline contours when viewed from above (bottom row).*

Weak Perspective projection

The pose-recovery method for orthographic projection can be simply extended for the case of weak perspective projection, when the field of view may be large but the object is still small. The most straightforward approach is to introduce a virtual orthographic image as in figure 7.5. The pose-recovery algorithm extended for weak perspective is

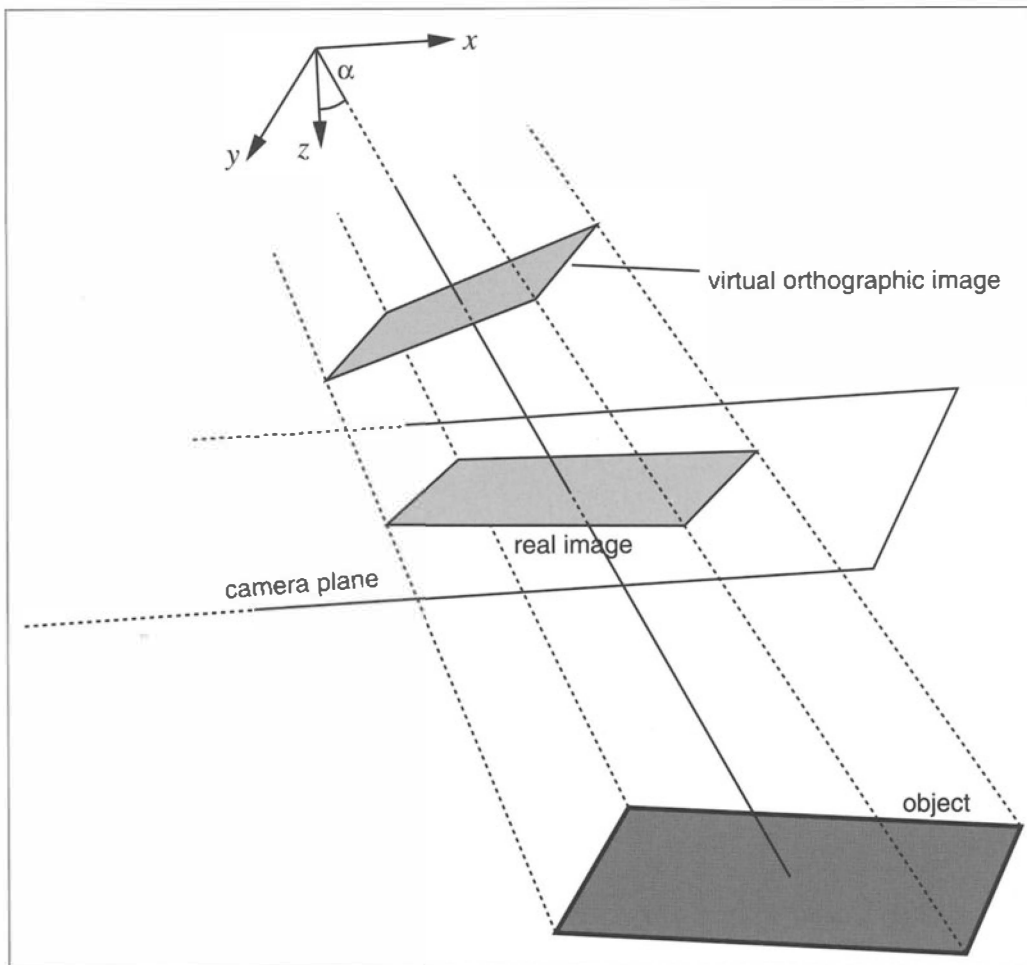


Figure 7.5: Pose correction for weak perspective. Under weak perspective, pose recovery proceeds as in the orthographic case, on a virtual orthographic image as shown.

1. Compute $\mathbf{u}$ and M from

$$\mathbf{X} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T.$$

2. Compute polar angles α, β :

$$\alpha = \arctan(|\mathbf{u}|/f), \quad \beta = \arctan(\mathbf{u}).$$

- 3.

$$M' = S_x(\sec \alpha) R(-\beta) M \quad \text{where} \quad S_x(\mu) \equiv \begin{pmatrix} \mu & 0 \\ 0 & 1 \end{pmatrix}$$

($R(-\beta)$ is represented here as a 2×2 matrix.)

4. Apply steps 2–7 of the orthographic algorithm of figure 7.1 to M' (in place of M), giving R', Z'_c (in place of R, Z_c).

- 5.

$$R = R_z(\beta) R_y(\alpha) R'.$$

(These rotation matrices are now all 3×3)

6. Translation:

$$\mathbf{R}_c = (X_c, Y_c, Z_c)^T \quad \text{where} \quad Z_c = Z'_c \cos \alpha, \quad X_c = u_1 Z_c / f, \quad Y_c = u_2 Z_c / f.$$

Figure 7.6: Recovery of pose of a plane under weak perspective.

outlined in figure 7.6. First, express the image-translation vector $\mathbf{u}$ in terms of polar angles

$$\alpha, \quad -\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2} \quad \text{and} \quad \beta, \quad -\pi \leq \beta \leq \pi,$$

that is, so that

$$\mathbf{u} = f \tan \alpha \begin{pmatrix} \cos \beta \\ \sin \beta \end{pmatrix}. \quad (7.2)$$

The next step 3 is to replace the affine matrix M by the affine matrix M' on the virtual orthographic plane. Then the orthographic pose-recovery algorithm is applied to M' in place of M , to give a three-dimensional rotation matrix R' (subject, of course, to reversal ambiguity) and translation in depth Z'_c , both in a virtual orthographic frame. Finally, these parameters are transformed back into the camera coordinate frame in steps 5 and 6.

7.2 Pose recovery for three-dimensional objects

The planar affine space survives intact as a subspace in the three-dimensional affine case (see (4.17) on page 90), so the pose-recovery method for the planar case continues to be usable, in principle, for the three-dimensional problem. Coefficients of the first 6 elements of the configuration vector $\mathbf{X}$ could be used to obtain $\mathbf{u}$, M as before and the planar pose-recovery algorithm would compute pose $\mathbf{R}_c$, R correctly, subject to the usual ambiguities. However, the planar algorithm does not make optimal use of the information now available. The additional pair of shape parameters contain valuable *parallax* information which is complementary to the information available in the planar case. The planar algorithm is at its least accurate close to the standard pose. The extreme case is the singularity at the standard pose itself, which gives rise to the ambiguity in orientation discussed earlier. In contrast, the additional parallax information is at its most useful precisely at the singularity, improving accuracy of recovered pose, and removing ambiguity, as figure 7.7 shows.

Once $\mathbf{Q}_0^z$ has been estimated for a given object (see below), the recovery of pose from some outline $\mathbf{X}$ in an image of the object is relatively straightforward — the algorithm is given in figure 7.8. Step 1 computes planar affine parameters $\mathbf{u}$, M and parallax parameter $\mathbf{v}$ from the shape-vector. Recovery of R , Z_c in steps 2 and 3 is based on solving (4.16) on page 88, expressing R in terms of its rows R_1, R_2, R_3 , which must be unit vectors, and mutually orthogonal. The matrix $\hat{R}$ resulting from step 3 need not be precisely orthogonal so step 4 finds the closest orthogonal matrix by “singular value decomposition” (appendix A.1) of $\hat{R}$. Finally, translation is calculated as in the planar affine case.

Figure 7.9 illustrates that the pose of a three-dimensional object can be recovered effectively over a wide range of views, using parallax as in the algorithm above.

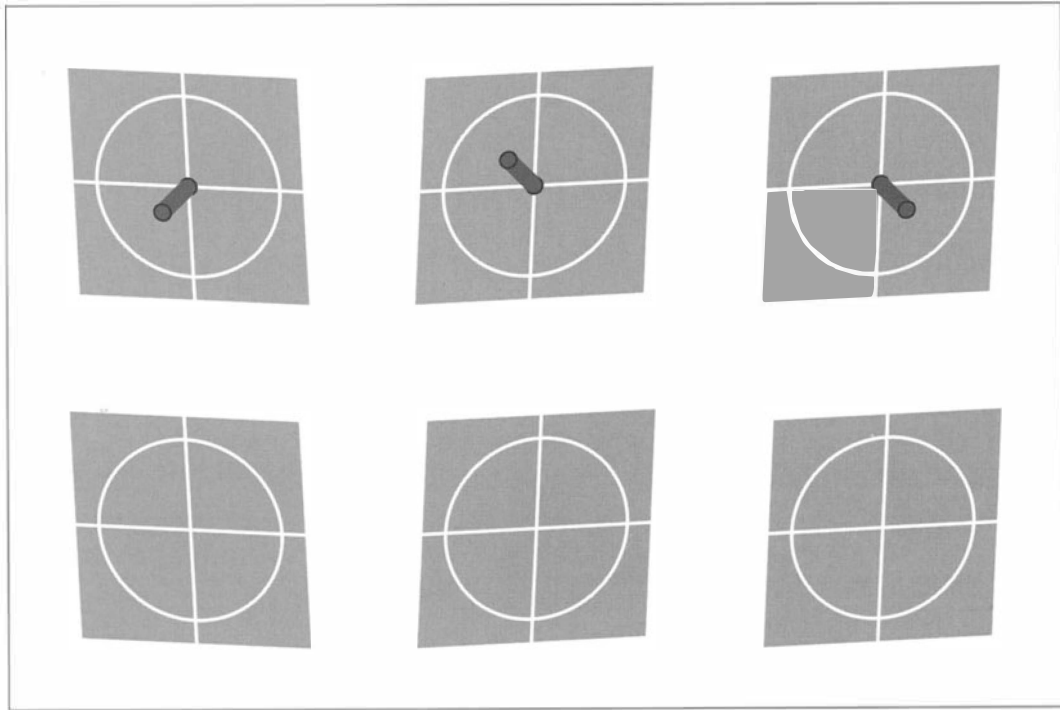


Figure 7.7: Parallax in pose recovery. *The poses of the top row of wheels, complete with axles, are easily discriminable. When parallax information is omitted by removing the axles, as on the bottom row, the orientations of the first two wheels are discriminable with difficulty and the last two actually appear identical because of the planar orientation ambiguity phenomenon.*

Estimating $\mathbf{Q}_0^z$

In principle, it would be possible to rotate the object, from its standard view, through 90° about the y -axis, and the new view would be

$$\begin{pmatrix} \mathbf{Q}_0^x \\ \mathbf{Q}_0^z \end{pmatrix}$$

from which $\mathbf{Q}_0^z$ could be obtained. This would be possible with a bent wire object, but with a solid object such as the leaves above only a modest angle of rotation is possible without alterations in hidden line structure. Therefore, practically, a method is required for estimating $\mathbf{Q}_0^z$ from a rotation R through a modest angle θ about an

1. Compute $\mathbf{u}$, M and $\mathbf{v}$ from the shape-vector $\mathbf{X}$, using:

$$\mathbf{X} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12}, v_1, v_2).$$

- 2.

$$Z_c = 2f [|(M_{11}, M_{12}, v_1)| + |(M_{21}, M_{22}, v_2)|]^{-1}$$

- 3.

$$\hat{R}_3 = \hat{R}_1 \times \hat{R}_2 \quad \text{where} \quad \begin{pmatrix} \hat{R}_1 \\ \hat{R}_2 \end{pmatrix} = \frac{Z_c}{f} (M \mid \mathbf{v}).$$

4. Enforce orthogonality:

$$R = UV \quad \text{where} \quad \hat{R} = UDV$$

(in which D should be approximately the identity I_3).

- 5.

Translation: $\mathbf{R}_c = (X_c, Y_c, Z_c)^T$ where $X_c = u_1 Z_c / f$, $Y_c = u_2 Z_c / f$.

Figure 7.8: Recovery of pose using parallax.

axis lying approximately parallel to the image plane. It is assumed that the object does not translate significantly in depth (Z) during the rotation.

The rotated view $\mathbf{r}_1(s) = U(s)\mathbf{Q}_1$ is a transformed version of standard views:

$$\mathbf{r}_1(s) = \frac{f}{Z_c} \left[\mathbf{u}_1 + R_{2 \times 3} \begin{pmatrix} X_0(s) \\ Y_0(s) \\ Z_0(s) \end{pmatrix} \right],$$

in which $Z_0(s)$ is as yet unknown. The translation $f\mathbf{u}_1/Z_c$ can be calculated simply as the centroid

$$\frac{f}{Z_c} \mathbf{u}_1 = \bar{\mathbf{r}}_1 = \begin{pmatrix} \mathbf{Q}_1^x \cdot \mathbf{1} \\ \mathbf{Q}_1^y \cdot \mathbf{1} \end{pmatrix}.$$

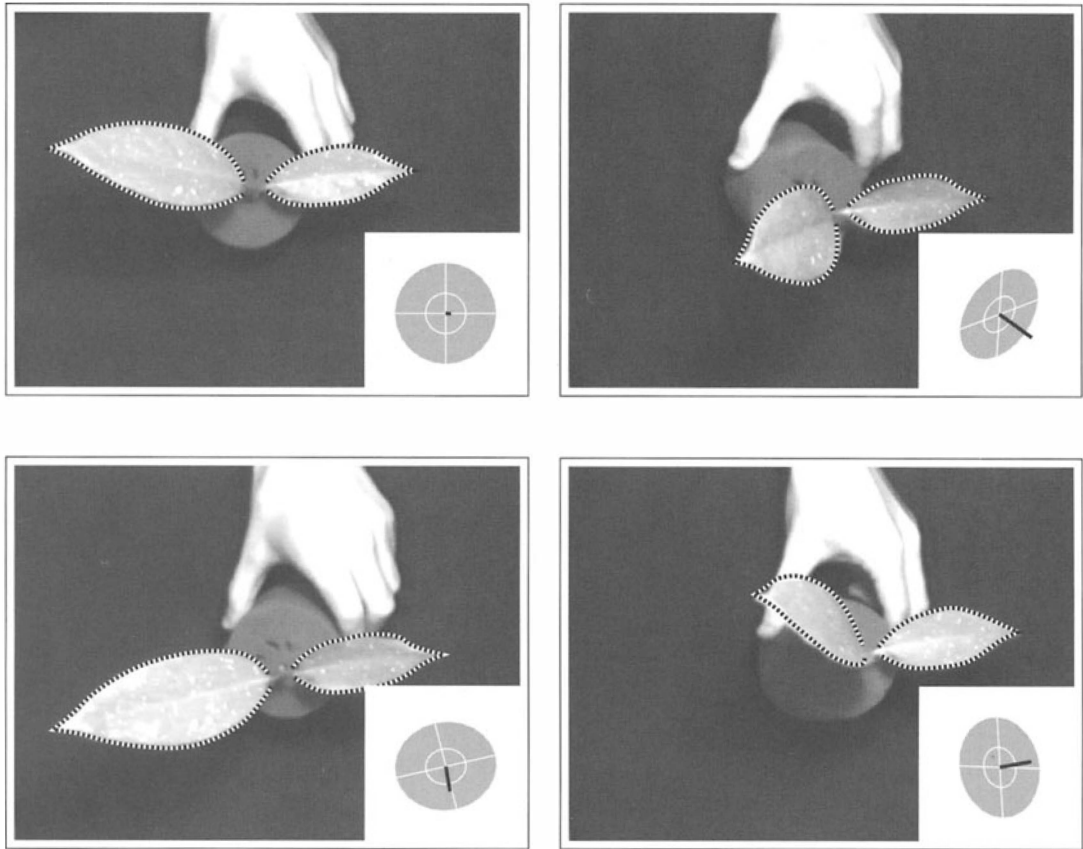


Figure 7.9: Pose recovery using the three-dimensional affine algorithm. *The pose can be accurately recovered over a wide range of transformations, including those near the planar affine degeneracy at the standard pose.*

Then $Z_0(s)$ can be estimated from

$$(\mathbf{a}^T \mathbf{a}) Z_0(s) = \mathbf{a}^T \left[\left(\frac{Z_c}{f} \mathbf{r}_1(s) - \mathbf{u}_1 \right) - R_{2 \times 2} \mathbf{r}_0(s) \right]$$

in which $\mathbf{a} = (R_{13}, R_{23})^T$. The resulting $Z_0(s)$ is a spline that can be expressed in terms of its control points as

$$\mathbf{Q}_0^z = \frac{1}{\mathbf{a}^T \mathbf{a}} (\mathbf{a}^T \otimes I) \left[\left(\frac{Z_c}{f} \mathbf{Q}_1 - \mathbf{u}_1 \otimes \mathbf{1} \right) - (R_{2 \times 2} \otimes I) \mathbf{Q}_0 \right] \quad (7.3)$$

where I is the $N_B \times N_B$ identity matrix and $\otimes$ is the “Kronecker product” operation (see appendix A.1).

A more accurate method

If the rotation matrix R in (7.3) is not accurately known, $\mathbf{Q}_0^z$ will be imperfectly estimated. The quality of the estimate can be improved by using more views and a more sophisticated algorithm. Suppose the views are $(\mathbf{Q}_n^x, \mathbf{Q}_n^y)$, $n = 1, \dots, N$, then the following simultaneous equations hold:

$$\begin{pmatrix} \mathbf{Q}_n^x \\ \mathbf{Q}_n^y \end{pmatrix} - \frac{f}{z_n} \left\{ \begin{pmatrix} x_n \mathbf{1} \\ y_n \mathbf{1} \end{pmatrix} + T_n \begin{pmatrix} \mathbf{Q}_0^x \\ \mathbf{Q}_0^y \\ \mathbf{Q}_0^z \end{pmatrix} \right\} = 0 \quad (7.4)$$

where (x_n, y_n, z_n, T_n) , $n = 1, \dots, N$ and $\mathbf{Q}_0^z$ are unknown. Here (x_n, y_n, z_n) is the object translation in view n and T_n is the 2×3 rotation matrix. If f is only known approximately, it can also be treated as an unknown in the system. The unknowns can be estimated using any non-linear minimisation method, for example conjugate gradient descent. Convergence is quick, and likely to find the correct local minimum, if the unknowns are initialised approximately. An alternative linear algorithm which does not rely on careful initialisation, is the “image-stream factorisation” algorithm of Tomasi and Kanade. This works by approximating the image stream, laid out as a matrix

$$\begin{pmatrix} \mathbf{Q}_1^x & \mathbf{Q}_1^y & \mathbf{Q}_2^x & \mathbf{Q}_2^y & \dots \end{pmatrix}$$

to have its theoretical rank (4 in this case), using singular value decomposition. Once this is done, $\mathbf{Q}_0^z$ could be calculated from the approximated image stream, subject to a certain fundamental affine indeterminacy.

7.3 Separation of rigid and non-rigid motion

In chapter 4 shape-spaces for combined rigid and non-rigid motion were constructed using key-frames. Such spaces are often too big for efficient or robust shape-fitting; they prove still to be useful for interpretation of shape, decomposing displacements into rigid and non-rigid components. This is done by projecting a fitted shape $\mathbf{Q}$ onto the shape-space and applying “Singular Value Decomposition” (SVD).

The decomposition algorithm is given in figure 7.10 (and justified below). The algorithm is illustrated in figure 7.11 for the problem of separating facial expression



Algorithm:

Given a fitted spline curve $\mathbf{Q}$:

1. Express $\mathbf{Q}$ in the key-frame basis:

$$\left(u_1, u_2, \{\hat{Y}_i^j, i = 0, 1, \dots, j = 1, 2, \dots\}\right) = W^+ \mathbf{Q},$$

where W is as defined in (4.21) on page 93.

2. Treating $\hat{Y}_i^j$ as components of a rectangular matrix $\hat{Y}$, apply SVD:

$$\hat{Y} = UDV.$$

3. Approximate Y as a rank 1 matrix

$$Y = UD^*V$$

where D^* is D with all but the largest singular value set to 0.

4. Recovered pose is given by the shape-vector

$$\mathbf{X} = (u_1, u_2, Y_0^1, Y_0^2, \dots)^T.$$

5. Recovered non-rigid deformation is expressed as a contour in standard pose:

$$\mathbf{Q}^d = \mathbf{Q}_0 + \sum_i \lambda_i \mathbf{Q}_i \quad \text{where} \quad \lambda_i = \frac{Y_i^1}{Y_0^1}, \quad i = 1, 2, \dots$$

and $\mathbf{Q}_i$ are key-frames as before.

Figure 7.10: Algorithm for decomposition of rigid and non-rigid displacement.

from head pose. Rigid transformations are modelled here as 3D affine, including parallax to accommodate the modest departure from co-planarity of tracked eyebrows, mouth and facial creases. Expression is represented in three-dimensional coordinates whose axes are defined by key-frames for smile, surprise and disgust, relative to a neutral expression.

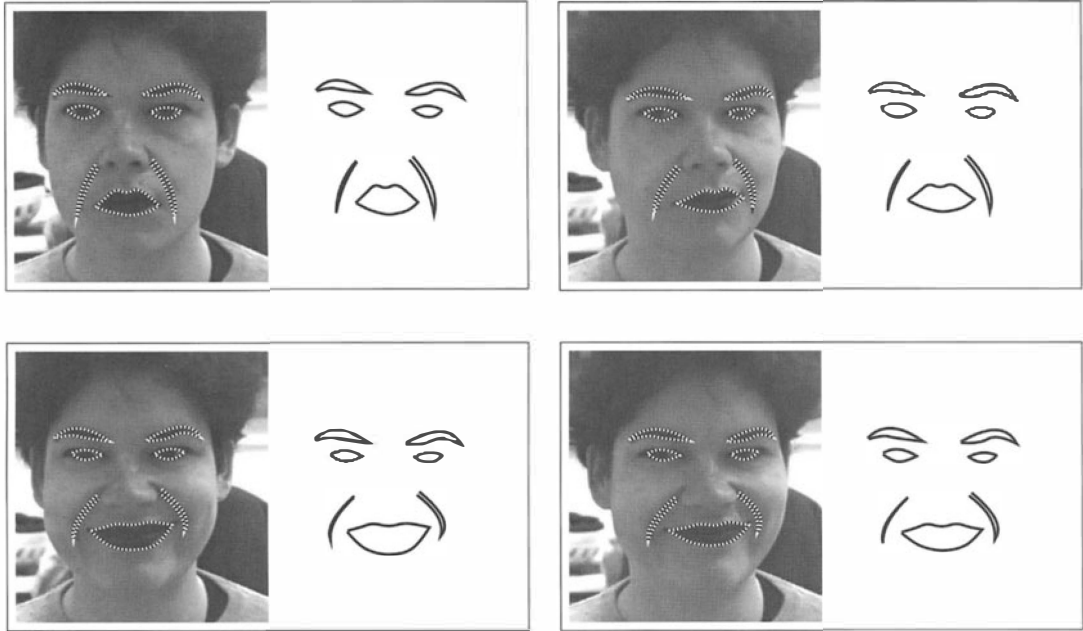


Figure 7.11: Pose-invariant transmission of facial expression. *Separation of non-rigid from rigid motion by SVD is used here to extract the facial expression of an actor. The extracted expression is displayed on this cat caricature in a fixed pose, and can be seen to be independent of the pose of the actor's head. (Figure courtesy of Benedicte Bascle.)*

Derivation of decomposition algorithm. Shape parameters Y_i^j , $i = 0, \dots, N_k$, $j = 1, \dots, N_r$ are highly interdependent, in principle, because, given the structure of shape-space in (4.21) on page 93, Y is a product of two vectors:

$$Y_i^j = \lambda_i X_{j+2},$$

where $\lambda_0 = 1$ is the weight of the template $\mathbf{Q}_0$ and λ_i is the weight of key-frame $\mathbf{Q}_i$ for $i = 1, \dots, N_k$. (The index $j+2$ is an offset to allow for two image-translation parameters.) Thus Y should have rank 1, and this is enforced by approximating $\hat{Y}$ to rank 1 in the SVD step. Note that one should expect D to contain one dominant singular value, indicating that Y is a good approximation to Y^* .

Bibliographic notes

Recovery of the pose of an object from its shape-space vector is based on an eigenvalue method for recovery of the pose of a textured surface in (Blake and Marinos, 1990). The “image-stream factorisation” algorithm (Tomasi and Kanade, 1991) uses singular value decomposition (SVD) (Barnett, 1990) of the image stream for motion analysis. Here, an adaptation of the algorithm was suggested for calibration of pose recovery using parallax. The algorithm for decomposition of rigid and non-rigid displacement is yet another application of SVD to bilinear decomposition problems in vision, others being structure and motion (Tomasi and Kanade, 1991), and shape and shading (Freeman and Tenenbaum, 1997).

Part II

Probabilistic Modelling

Chapter 8

Probabilistic models of shape

The purpose of this second part of the book is to put Active Contours into a probabilistic setting. As chapter 2 claimed, the probabilistic framework is essential for dealing with *classes* of shapes and motions. It is valuable even with deformable templates, in static problems, to describe classes of shapes. Then probabilistic modelling is extended to dynamic problems, to mesh with the powerful Kalman filtering formalism, in which cumulative temporal uncertainty about shape is counterbalanced by the inflow of measurements from an image sequence.

This chapter concentrates on the application of probabilistic models to static problems. The ideas discussed so far about fitting curves by regularisation are to be re-interpreted probabilistically. The deterministic approach of chapter 6 aimed to generate a unique estimate $\hat{\mathbf{X}}$ of curve shape from data $\mathbf{r}_f(s)$, moderated via regularisation towards a template $\overline{\mathbf{X}}$. Now, in a more general probabilistic setting, $\hat{\mathbf{X}}$ is merely one property, typically the mean or mode, of an entire probability distribution. The solution to the fitting problem is therefore no longer just a single value, but a whole family of possible curves. In that case, the variance of the distribution is a measure of how accurate the fitted curve is and can be used to generate a range of plausible fitted curves. One application for this is to sweep out the search region for image-processing operations. Another application, developed in later chapters on tracking, is to achieve a fusion of shape information accumulated over time and the latest visual measurements.

The distribution for curve shape $\mathbf{X}$ obtained from probabilistic fitting is expressed as a *posterior density* $p(\mathbf{X}|\mathbf{r}_f)$. This is the conditional probability density for the curve

shape $\mathbf{X}$ *given* the observed data $\mathbf{r}_f(s)$. According to Bayes' formula for densities (see appendix A.3) the posterior density can be obtained as a product of a *prior density* $p_0(\mathbf{X})$ and an *observation density* $p(\mathbf{r}_f|\mathbf{X})$:

$$p(\mathbf{X}|\mathbf{r}_f) \propto p(\mathbf{r}_f|\mathbf{X})p_0(\mathbf{X}). \quad (8.1)$$

Note that although $p(\mathbf{r}_f|\mathbf{X})$ is a probability density over $\mathbf{r}_f$, in this formula $\mathbf{r}_f$ is considered fixed and it is the variation of $p(\mathbf{r}_f|\mathbf{X})$ with $\mathbf{X}$ — the “likelihood” of $\mathbf{X}$ — that is of interest. The prior density is the probabilistic mechanism for regularisation. For example, the norm-squared regulariser $\alpha\|\mathbf{X} - \bar{\mathbf{X}}\|^2$ used in chapter 6 becomes a Gaussian density function $p(\mathbf{X})$ whose mean is the curve $\bar{\mathbf{X}}$. The other term $\|\mathbf{r} - \mathbf{r}_f\|^2$ in the regularisation problem, conveying the influence of the data, also becomes a Gaussian density whose value is high when the hypothesised curve $\mathbf{r}$ fits the data closely. Then Bayes' rule simply combines the competing influences of the prior and the observations into a single density, also Gaussian.

Note that Bayesian principles of image interpretation reach far beyond Gaussian modelling. However, the Gaussian is the simplest case and has attractive properties that facilitate efficient computation. Consideration of more general distributions is left until much later in the book, in chapter 12.

8.1 Probability distributions over curves

The Gaussian prior probability density for curves in shape-space $\mathcal{S}$ consistent with the general quadratic regulariser $(\mathbf{X} - \bar{\mathbf{X}})^T \bar{\mathcal{S}} (\mathbf{X} - \bar{\mathbf{X}})$ is:

$$p_0(\mathbf{X}) \propto \exp -\frac{1}{2}(\mathbf{X} - \bar{\mathbf{X}})^T \bar{\mathcal{S}} (\mathbf{X} - \bar{\mathbf{X}}), \quad (8.2)$$

where the matrix $\bar{\mathcal{S}}$ is the “information” matrix introduced in chapter 6. Note that probability density decreases as the curve $\mathbf{X}$ deviates further from the mean $\bar{\mathbf{X}}$ of the distribution, as expected. Probabilistically, $\bar{\mathcal{S}}$ has a particular interpretation: its inverse $P_0 = \bar{\mathcal{S}}^{-1}$, if it exists, is a *covariance matrix* (appendix A.3), a multi-dimensional measure of the variability of curve shape across a distribution. It can be used to compute the positional variability of a given point $\mathbf{r}(s)$ on a curve as a 2×2 covariance matrix

$$P_{\mathbf{r}}(s) = U(s)W P_0 W^T U(s)^T, \quad (8.3)$$

in a two-dimensional Gaussian distribution $\mathcal{N}(\bar{\mathbf{r}}(s), P_{\mathbf{r}}(s))$. In general, this distribution can be depicted as an ellipse whose axes are eigenvectors of $P_{\mathbf{r}}(s)$ and whose semi-axes, representing positional variances, are the eigenvalues. The spatial variance $P_{\mathbf{r}}(s)$ is represented by an uncertainty ellipse giving a mean-square displacement

$$\rho_0^2(s) = \text{tr}(P_{\mathbf{r}}(s)) \quad (8.4)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix. The mean-square displacement along the entire curve can then be computed easily (the proof follows below) as

$$\bar{\rho}_0^2 = \text{tr}(P_0 \mathcal{H}). \quad (8.5)$$

An isotropic prior, uniform in s , with mean-square displacement along the curve $\bar{\rho}_0^2$ is projected onto shape-space $\mathcal{S}$ as

$$p_0(\mathbf{X}) \propto \exp - \frac{N_X}{2\bar{\rho}_0^2} \|\mathbf{X} - \bar{\mathbf{X}}\|^2.$$

This can be seen by considering its information matrix in shape-space, which is

$$\bar{S} = \frac{N_X}{\bar{\rho}_0^2} \mathcal{H}. \quad (8.6)$$

From (8.5) it is apparent that the mean-square displacement along the curve is

$$\text{tr} \left(\frac{\bar{\rho}_0^2}{N_X} \mathcal{H}^{-1} \mathcal{H} \right) = \frac{\bar{\rho}_0^2}{N_X} \text{tr}(I_{N_X}) = \bar{\rho}_0^2,$$

as required.

Spline space is an interesting special case in which the covariance at a given point $\mathbf{r}(s)$ is $P_{\mathbf{r}}(s) \propto I_2$, a multiple of the identity matrix, so that positional error at each point is isotropically distributed (see below for the derivation). The average displacement of $\mathbf{r}(s)$ from its mean $\bar{\mathbf{r}}(s)$, is $\rho_0(s)$ and in fact $\rho_0(s) \approx \bar{\rho}_0$ for all s . This suggests circular confidence regions for $\mathbf{r}(s)$ (figure 8.1). (The average pointwise displacement $\rho_0(s)$ varies a little with s , relative to $\bar{\rho}_0$ — by around $\pm 10\%$, for instance, on a regular, closed, quadratic B-spline.)

In chapter 6, there was some discussion of the way in which image-processing operations are applied along curve normals. In the interests of efficiency, processing is limited to a segment of each normal within a search region. The probabilistic

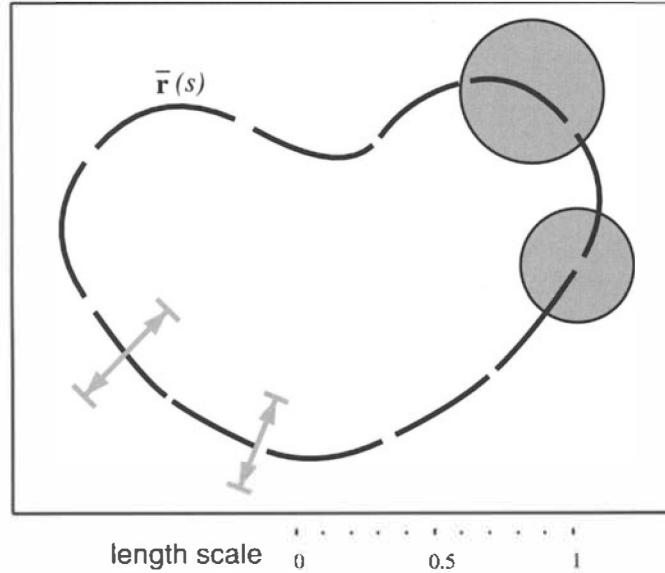


Figure 8.1: Probability distribution for curves. *An example of a Gaussian probability distribution in spline space with mean shape $\bar{\mathbf{r}}(s)$ and uniform, isotropic covariance P_0 . The distribution of the position of a given point $\mathbf{r}(s)$ is Gaussian, with circular confidence intervals of radius $1.73\rho_0(s)$ (95% confidence). The normal displacement has a Gaussian distribution with standard deviation $\rho_n(s)$ and $\rho_n(s) = \rho_0(s)/\sqrt{2}$ in the special case of this example. The search segments shown (arrowed) are intervals of length $\pm 2\rho_n(s)$ for approximately 95% statistical confidence. (Average displacement $\bar{\rho}_0 = 0.14$ here.)*

framework provides a rationale, using the prior distribution, for fixing the search segment along the normal $\bar{\mathbf{n}}(s)$ at $\bar{\mathbf{r}}(s)$. The normal component of displacement of the curve is Gaussian with mean 0 and standard deviation $\rho_n(s)$ where

$$\rho_n^2(s) = \bar{\mathbf{n}}(s)^T P_{\mathbf{r}}(s) \bar{\mathbf{n}}(s).$$

or, directly in terms of shape-space covariance P_0 ,

$$\rho_n^2(s) = \mathbf{h}(s)^T P_0 \mathbf{h}(s) \quad (8.7)$$

where $\mathbf{h}(s)$ was defined in (6.15) on page 124 in chapter 6. In the special case of the norm-squared prior ($S \propto \mathcal{H}$) in spline space, the isotropy of $P_{\mathbf{r}}(s)$ means that $\rho_n(s) = \rho_0(s)/\sqrt{2}$.

A reasonable search segment is $\bar{\mathbf{r}}(s) \pm 2\rho_n(s)\bar{\mathbf{n}}(s)$, corresponding to a confidence interval (for the one-dimensional Gaussian distribution) at a level of approximately 95%. Such search segments are illustrated in figure 8.1. This idea is elaborated later for validation gates, taking account not only of the prior but also the observation density.



Derivation of average radius and isotropy results

The formula (8.5) for the root-mean-square displacement for the covariance ellipse is derived first. Pointwise, the mean-square displacement is $\rho_0^2(s)$ whose average value along the curve, from (8.4), is

$$\begin{aligned}
 & \frac{1}{L} \int_0^L \text{tr}(P_{\mathbf{r}}(s)) \, ds \\
 \stackrel{(8.3)}{=} & \frac{1}{L} \int_0^L \text{tr}(U(s)W P_0 W^T U(s)^T) \, ds \\
 = & \text{tr} \left(P_0 W^T \left[\frac{1}{L} \int_0^L U(s)^T U(s) \, ds \right] W \right) \\
 \stackrel{(3.23)}{=} & \text{tr}(P_0 W^T \mathcal{U} W) \\
 \stackrel{(4.7)}{=} & \text{tr}(P_0 \mathcal{H}),
 \end{aligned}$$

as required.

The special case of the norm-squared prior (8.6) in spline space $\mathcal{S} = \mathcal{S}_Q$ was considered in which $W = I_{N_Q}$ and the pointwise covariance is

$$P_{\mathbf{r}}(s) = U(s)P_0 U(s)^T.$$

Since $P_0 \propto \mathcal{U}^{-1}$,

$$P_{\mathbf{r}}(s) \propto (\mathbf{B}(s)^T \mathcal{B}^{-1} \mathbf{B}(s)) I_2$$

— a multiple of the identity matrix as claimed.



Random sampling

A graphic illustration of a statistical family of curves can be made by sampling randomly from its distribution. Random curve sampling is a practical technique in its own right, partly as a debugging aid to check whether a particular probabilistic prior model of shape is appropriate to a given application. More importantly it forms the

basis of some powerful algorithms for recognising patterns, used when the posterior density for shape given data is too complex to be represented exactly and is represented instead by a set of samples.

Given an N_X -dimensional Gaussian distribution $\mathcal{N}(\bar{\mathbf{X}}, P)$ in shape-space with mean $\bar{\mathbf{X}}$ and covariance matrix P , a random variate $\mathbf{X}$ from the distribution can be generated by taking a vector $\mathbf{w}$ of N_X independent “standard” normal variables each distributed as $\mathcal{N}(0, 1)$ and transforming $\mathbf{w}$ linearly:

$$\mathbf{X} = B\mathbf{w} + \bar{\mathbf{X}} \quad (8.8)$$

where

$$B = \sqrt{P}, \quad (8.9)$$

a matrix square root with the property that $BB^T = P$ (Note that this does not uniquely specify B , but any admissible B serves equally well, and generates the desired distribution; see also appendix A.1). This technique is used to illustrate prior distributions of curves in spline space $\mathcal{S}_Q$ (figure 8.2) and in various shape-spaces (figure 8.3). As before, the distributions are given by norm-squared densities over shape-space.

8.2 Posterior distribution

The aim of the chapter is to express curve fitting in terms of the posterior distribution for shape $\mathbf{X}$ given observed data $\mathbf{r}_f$. Realistically, $\mathbf{r}_f$ must be obtained by processing image data, and that is addressed in the next section. For now, the posterior is illustrated with artificial data, using the random sampling method. As at the start of chapter 6, the artificial image feature is modelled as a spline curve with control-vector $\mathbf{Q}_f$. What was characterised there, in (6.3) on page 117, as fitting by regularisation, becomes the construction by Bayes’ formula (8.1) of the posterior distribution $p(\mathbf{X}|\mathbf{Q}_f)$, given a Gaussian prior $p_0(\mathbf{X})$ and an observation density $p(\mathbf{Q}_f|\mathbf{X})$. The measurement term $\|\mathbf{Q} - \mathbf{Q}_f\|^2$ in the regularisation problem is interpreted as an observation density

$$p(\mathbf{Q}_f|\mathbf{X}) \propto \exp - \frac{N_X}{2\bar{\rho}_f^2} \|\mathbf{Q} - \mathbf{Q}_f\|^2 \quad (8.10)$$

where a variance constant $\bar{\rho}_f^2/N_X$ has been included here to represent explicitly the uncertainty of measurements. The distance constant $\bar{\rho}_f$ can be interpreted as the average displacement, at a given point on the fitted curve, due solely to observation error and without any influence from the prior. Finally, multiplying prior and observation densities, in accordance with Bayes' formula, the posterior distribution proves to be a Gaussian

$$\mathbf{X}|\mathbf{Q}_f \sim \mathcal{N}(\hat{\mathbf{X}}, P). \quad (8.11)$$

Its mean $\hat{\mathbf{X}}$ is precisely the solution (6.7) to the earlier regularisation problem, adjusted now to allow for the observation variance constant:

$$\hat{\mathbf{X}} = S^{-1} \left(\bar{S} \bar{\mathbf{X}} + \frac{N_X}{\bar{\rho}_f^2} \mathcal{H} \mathbf{X}_f \right) \quad (8.12)$$

where

$$S = \bar{S} + \frac{N_X}{\bar{\rho}_f^2} \mathcal{H}. \quad (8.13)$$



Figure 8.2: Sampling from curve families. Samples are drawn at random (left) from a uniform, isotropic, Gaussian distribution in the spline space S_Q such that $\|\mathbf{r}(s) - \bar{\mathbf{r}}(s)\|$ has a root-mean-square value of 0.2 length units. (Mean shape $\bar{\mathbf{r}}$ is shown dashed). Parallel curves depicting confidence intervals for normal displacement (spatially averaged), at 95% significance, contain the random curves as expected (right).

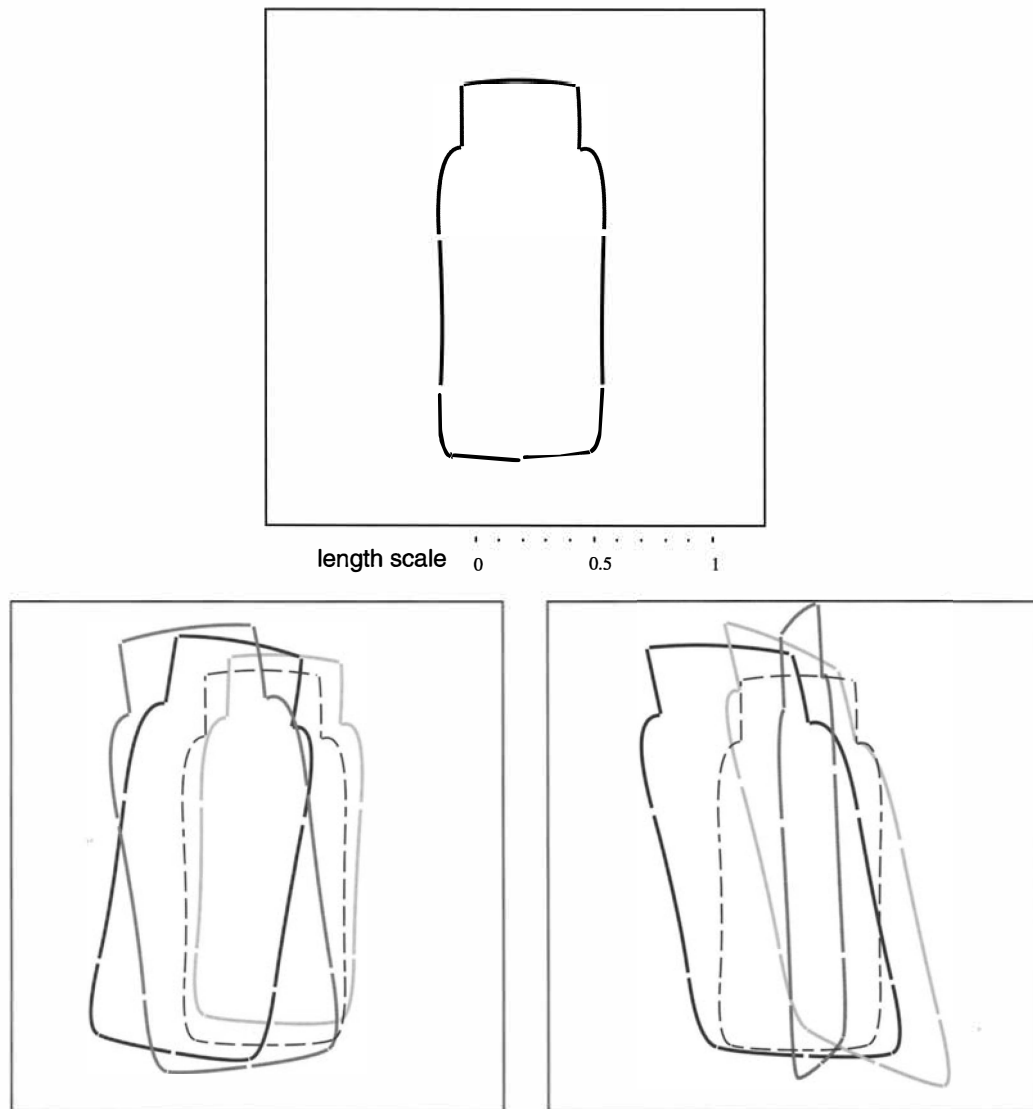


Figure 8.3: Random sampling in shape-space *The families of bottle-outlines shown have mean shape (top) and a uniform, isotropic, Gaussian distribution in shape-space with root-mean-square displacement of 0.3 length units. Euclidean similarities (left); affine space (right).*

The estimate $\hat{\mathbf{X}}$ is also known as a “Maximum A Posteriori” (MAP) estimate because it is the value of $\mathbf{X}$ at which the posterior probability density $p(\mathbf{X}|\mathbf{Q}_f)$ achieves its peak value. The covariance of the posterior distribution is $P = S^{-1}$, and S , the total information matrix for the posterior, is just the sum of information in the prior and observation distributions. (The derivation of the posterior distribution is essentially a replay of the derivation on page 118 of the solution to the regularisation problem.)

The posterior distribution for the curve-fitting problem of figure 6.2 on page 119 can be illustrated by random sampling. Again, regularisation is made invariant to transformations in a subspace $\mathcal{S}_s$. As in chapter 6, this is represented probabilistically by modifying the information matrix (8.6) for a uniform, isotropic, Gaussian prior using a projection E^d outside the invariant subspace:

$$\bar{S} = \frac{N_X}{\bar{\rho}_0^2} E^{dT} \mathcal{H} E^d. \quad (8.14)$$

This has the effect of producing a prior distribution that is invariant to Euclidean similarities so that

$$p_0(\mathbf{X} + \mathbf{X}^s) = p_0(\mathbf{X}) \quad \text{for any } \mathbf{X}^s \in \mathcal{S}_s.$$

The posterior distribution, illustrated by random sampling in figure 8.4, clearly shows the likely range of variability of possible solutions, something that regularisation alone could not convey. This is also reflected mathematically by the fact that there is an extra constant in the system. As before there is a regularisation constant α which is now a ratio of the coefficients of the information matrices for the observation and for the prior:

$$\alpha = \bar{\rho}_f^2 / \bar{\rho}_0^2$$

and which determines the mean $\hat{\mathbf{X}}$ of the posterior distribution. In addition there is now also the spatial variance parameter $\bar{\rho}_0^2$ for the prior which also affects the variability of fitted curves as modelled by the posterior distribution.

Finally, exploiting the additivity of information in (8.13), a mean spatial variance $\bar{\rho}^2$ for the posterior can be computed from

$$\frac{1}{\bar{\rho}^2} = \frac{1}{\bar{\rho}_0^2} + \frac{1}{\bar{\rho}_f^2}$$

and is exact if there is no invariant subspace ($\mathcal{S}_s = 0$), and otherwise is a lower bound on mean variance. A confidence region based on $\bar{\rho}$ is shown in figure 8.4.

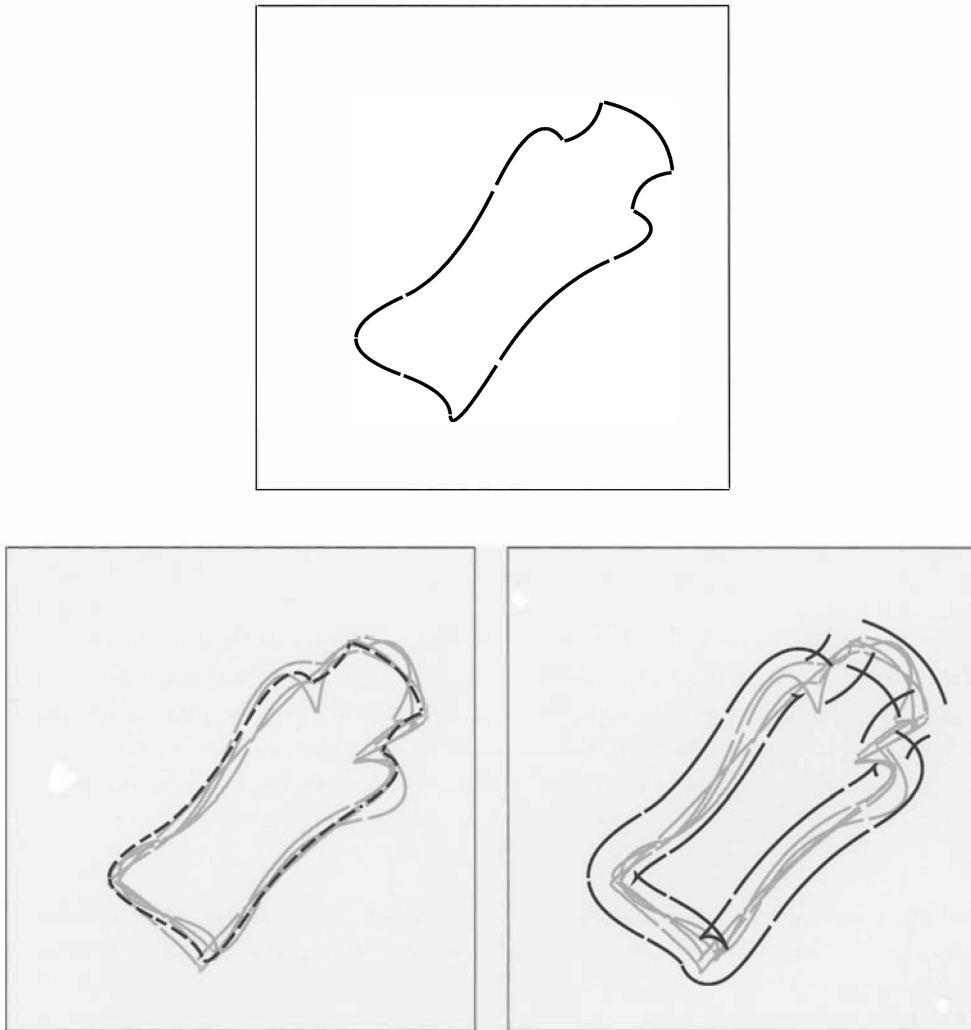


Figure 8.4: Sampling from the posterior *Data (top) is given, together with a prior over spline space with the Euclidean similarities as an invariant subspace (to allow free rotation and translation). Curves sampled from the posterior distribution sweep out a distribution (left), whose mean (dashed line) is simply the regularised estimate obtained in chapter 6 (figure 6.2 on page 119), with $\alpha = 0.5$. The sampled curves (right) lie comfortably inside a 95% confidence interval.*

Note that the matrix $\bar{S}$ in (8.14) has become singular, so that the covariance matrix P_0 does not exist and the prior density p_0 cannot formally be normalised. In fact p_0 can still be treated as a consistent Gaussian distribution. It consists of a valid Gaussian density restricted to shape-space $\mathcal{S} \ominus \mathcal{S}_s$ excluding the invariant subspace $\mathcal{S}_s$, together with a uniform density over $\mathcal{S}_s$. It is the uniform density that cannot be normalised and gives rise to unbounded covariance. We could treat it formally as a sequence of Gaussians of increasing variance, understanding that any expression incorporating p_0 will be evaluated in the limit, where one exists. In fact this is achieved for our purposes simply by applying additivity of information to $\bar{S}$ just in the same way as for a non-singular $\bar{S}$. The result is a *posterior* Gaussian whose covariance generally exists and which can be normalised.



8.3 Probabilistic modelling of image features

The example above of a posterior distribution is a contrived one, for tutorial purposes, driven by ideal data in the form of a spline curve $\mathbf{Q}_f$. In chapter 6, methods were developed for fitting to real data $\mathbf{r}_f$ in the form of a curve sampled from an image. The importance of using only the normal component of displacement was explained. Now the probabilistic approach needs to be developed to encompass normal displacements in real image data.

As in the previous section, the data-term in the regularisation problem can be interpreted probabilistically, but now based on normal displacement to give an observation density:

$$p(\mathbf{r}_f|\mathbf{X}) \propto \exp - \frac{N_X}{2\bar{\rho}_f^2} \|\mathbf{r} - \mathbf{r}_f\|_{\bar{\mathbf{n}}}^2 \quad \text{with} \quad \mathbf{r}(s) = U(s)(W\mathbf{X} + \mathbf{Q}_0). \quad (8.15)$$

It can be shown (see below) that $\bar{\rho}_f^2$ is the contribution to mean-square value of the normal displacement $(\mathbf{r} - \bar{\mathbf{r}}) \cdot \bar{\mathbf{n}}$, in the fitted curve, due solely to measurement error.

Now the discrete form (6.13) from chapter 6 for $\|\mathbf{r} - \mathbf{r}_f\|_{\bar{\mathbf{n}}}^2$ can be used, giving approximately:

$$p(\mathbf{r}_f|\mathbf{X}) \propto \exp - \frac{1}{2\sigma^2} \sum_{i=1}^N [(\mathbf{r}_f(s_i) - \mathbf{r}(s_i)) \cdot \bar{\mathbf{n}}(s_i)]^2. \quad (8.16)$$

where

$$\sigma = \bar{\rho}_f \sqrt{\frac{N}{N_X}}. \quad (8.17)$$

One interpretation of this observation density is in terms of the variability of individual measurements. Suppose the marginal density of a single measurement

$$[\mathbf{r}_f(s_i) - \mathbf{r}(s_i)] \cdot \bar{\mathbf{n}}(s_i)$$

were a Gaussian $\mathcal{N}(0, \sigma^2)$ variable. Then the joint density (8.16) could be interpreted as treating the data $\mathbf{r}_f$ as a series of sampled image features

$$\mathbf{r}_f(s_i) \sim \mathcal{N}(\mathbf{r}(s_i), \sigma^2) \quad (8.18)$$

with mutually *independent* Gaussian distributions. To some extent such independence is plausible but it depends rather on what is the chief source of the variability in image-feature measurements. If the variability were thought to derive from noise, for example electrical or optical noise at individual pixels, it is probably defensible to regard noise sources at neighbouring pixels as independent; even then, noise due to lighting flicker (especially fluorescent lighting) or power supply ripple would be highly correlated across pixels. In practice, however, the variability in feature position due to such noise is negligible. The dominant sources of variability reflect grosser discrepancies than mere device noise from the camera.

- Imperfections in the shape-space model mean that the image outline $\mathbf{r}_f$ will not lie entirely in the shape-space $\mathcal{S}$. The difference

$$\mathbf{e}(s) = \mathbf{r}_f(s) - U(s)(WX_f + \mathbf{Q}_0)$$

between the outline and its projection onto shape-space acts as spatially correlated noise.

- Background texture and occasional obscuration by passing objects will cause features to be picked up by image processing that do not belong to the modelled object. Such errors may be gross and, again, spatially correlated.

The result is that statistical independence between neighbouring $\mathbf{r}_f(s_i) | \mathbf{X}$ is plausible only if features are not sampled too densely, and this is a limitation of the observation model (8.15). It is therefore important to note that the interpretation of (8.16) in terms of statistically independent measurements $\mathbf{r}_f(s_i)$ is by no means the only consistent interpretation (see discussion at the end of this section).

Algorithm to construct the posterior

Construction of the posterior density $p(\mathbf{X}|\mathbf{r}_f)$ from image measurements $\mathbf{r}_f(s_i)$, $i = 1, \dots, N$ uses precisely the recursive fitting algorithm from chapter 6 (figure 6.7 on page 127), but now interpreted probabilistically. The algorithm computes an “aggregated observation” $\mathbf{Z}$ and the associated statistical information for estimation of $\mathbf{X}$ is S , also output by the algorithm. In fact

$$\mathbf{Z}|\mathbf{X} \sim \mathcal{N}(S\mathbf{X}, S),$$

so that $\mathbf{Z}$ is an unbiased estimator for $S\mathbf{X}$ (rather than for $\mathbf{X}$ directly). Its covariance is in fact S (*sic*), following from the fact that the statistical information in $S^{-1}\mathbf{Z}$ is S .

The posterior distribution is a Gaussian

$$\mathbf{X}|\mathbf{Z} \sim \mathcal{N}(\hat{\mathbf{X}}, P) \quad \text{where } P = (\bar{S} + S)^{-1}. \quad (8.19)$$

An example of a posterior constructed in this way from an image is given in figure 8.5.

Independence of measurements

The functional form of the density $p(\mathbf{r}_f|\mathbf{X})$ does not in fact uniquely determine the conditional distribution for observations until a parameterisation is specified for $\mathbf{r}_f(s)$, the image feature. For example $\mathbf{r}_f(s)$ could be modelled as a spline with parameters $\mathbf{Q}_f$ of dimension N_f . Then the conditional density can be used to compute probabilities via integrals of the form

$$\int p(\mathbf{r}_f|\mathbf{X}) d\mathbf{Q}_f.$$

To examine the question of independence of the $\mathbf{r}_f(s_i)$, the conditional density must be transformed from the $\mathbf{Q}_f$ parameterisation to a new parameter set $(\mathbf{r}_f(s_1), \dots, \mathbf{r}_f(s_N))$. This could be done (without loss of generality) by expressing the new parameters in terms of the first N components in the vector $\mathbf{Q}_f$, and integrating over the remaining $N_f - N$ components. This is possible *only if* $N < N_f$. Otherwise there exists no density in the new variables, let alone an independent one. This is intuitively reasonable. When the underlying parameterisation $\mathbf{Q}_f$ of the image curve has relatively few degrees of freedom (N_f is small), this represents an image curve with strong spatial correlation, such as a spline with few control points. In that case, successive measurements made on the curve cannot be mutually independent. To summarise, the form of (8.16) does not force the deduction that individual image measurements $\mathbf{r}_f(s_i)$ are independent. This conclusion is important to the operation of the validation gate, discussed below.



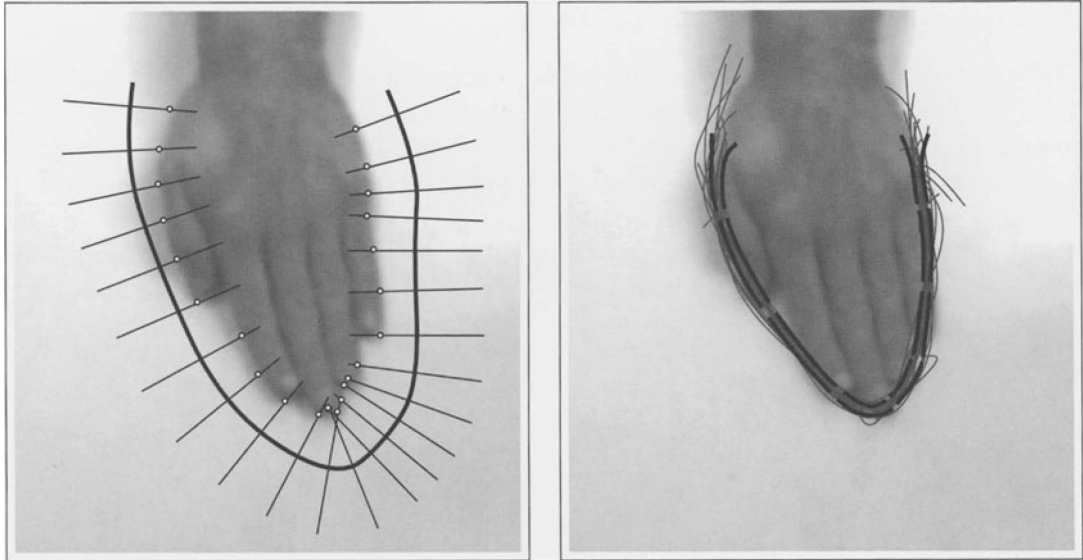


Figure 8.5: Constructing a posterior distribution from an image. *The image data (left) is shown overlaid with a broad search region determined by the chosen prior. When combined with image observations, a relatively tight posterior distribution results, illustrated here by randomly sampled curves and 95% confidence region (right).*

Mean-square normal displacement

It was claimed above that $\bar{\rho}_f^2$ is the mean-square value of the normal displacement $(\mathbf{r} - \mathbf{r}_f) \cdot \mathbf{n}$. This could be proved directly by integration, as was done earlier for $\bar{\rho}_0$ in the norm-squared prior. A concise alternative is to appeal to the well-known property of Boltzmann distributions in statistical mechanics for random systems whose energy $U(\mathbf{X})$ is a quadratic function of parameters $\mathbf{X}$ and is distributed as $p(\mathbf{X}) \propto \exp -U$. The mean energy of such a system is simply $\bar{U} = N_X/2$, where N_X is the number of degrees of freedom of the parameter set.



8.4 Validation gate

The validation gate was introduced in chapter 6 (section 6.3) as a mechanism for detecting outliers, features that are misplaced or missing when, for instance, part of an object is obscured. The idea was to test the innovation ν_i to ensure that it was not too large; the difficulty was to decide how large is too large. Now in the probabilistic

framework, there is a basis for making this test as a statistical hypothesis test on the innovation, whose Gaussian distribution is known. The innovation (6.14) can be expanded as

$$\nu_i = [\mathbf{r}(s_i) - \bar{\mathbf{r}}(s_i)] \cdot \bar{\mathbf{n}}(s_i) + [\mathbf{r}_f(s_i) - \mathbf{r}(s_i)] \cdot \bar{\mathbf{n}}(s_i),$$

in which the first term has a $\mathcal{N}(0, \rho_n^2(s))$, distribution, from (8.7). The second term is a little more difficult to deal with. In the case that the measurements $\mathbf{r}_f(s_i)$ are considered independent, it is Gaussian $\mathcal{N}(0, \sigma^2)$ from (8.18). At the other extreme, in the limit of high spatial interdependence of measurements, its variance should tend towards 0. This means that

$$\nu_i \sim \mathcal{N}(0, \rho^2(s)) \text{ where } \rho_n^2(s) \leq \rho^2(s) \leq \rho_n^2(s) + \sigma^2. \quad (8.20)$$

A conservative assumption of high spatial dependence would suggest

$$\rho(s) = \rho_n(s). \quad (8.21)$$

The distribution of the innovation suggests a test that if $|\nu_i| > \kappa\rho(s)$ then the measurement $\mathbf{r}_f(s)$ is invalidated. Taking the default value $\kappa = 2$, at approximately 95% statistical significance the measurement is regarded as an outlier. In practice this is implemented by using a search segment (figure 8.1) of length $\rho(s)$ so that when $|\nu_i| > 2\rho(s)$, no feature will be found. The recursive fitting algorithm of chapter 6 can be modified to take account of outliers. The iterative step in the original algorithm (figure 6.7 on page 127) is rewritten to ignore outliers, as in figure 8.6.

Note that the modified algorithm requires that $\bar{S}$ have full rank and that excludes the use of an invariant subspace in the prior which always makes $\bar{S}$ singular. This is a real and reasonable limitation. The idea of the invariant subspace is that it allows certain shape variables to be unconstrained. In that case, the absolute position of a given point $\mathbf{r}(s)$ on the curve is also unconstrained by the prior (even though constraints may apply to the relative positions of pairs of points). Validity checks are only effective when the constraints imposed by the prior on individual points are reasonably tight. This requires the prior on the shape $\mathbf{X}$ as a whole to be reasonably tight — none of the eigenvalues of $\bar{S}$ should be too close to 0. The effect of outlier rejection on fitting is illustrated in figure 8.7.

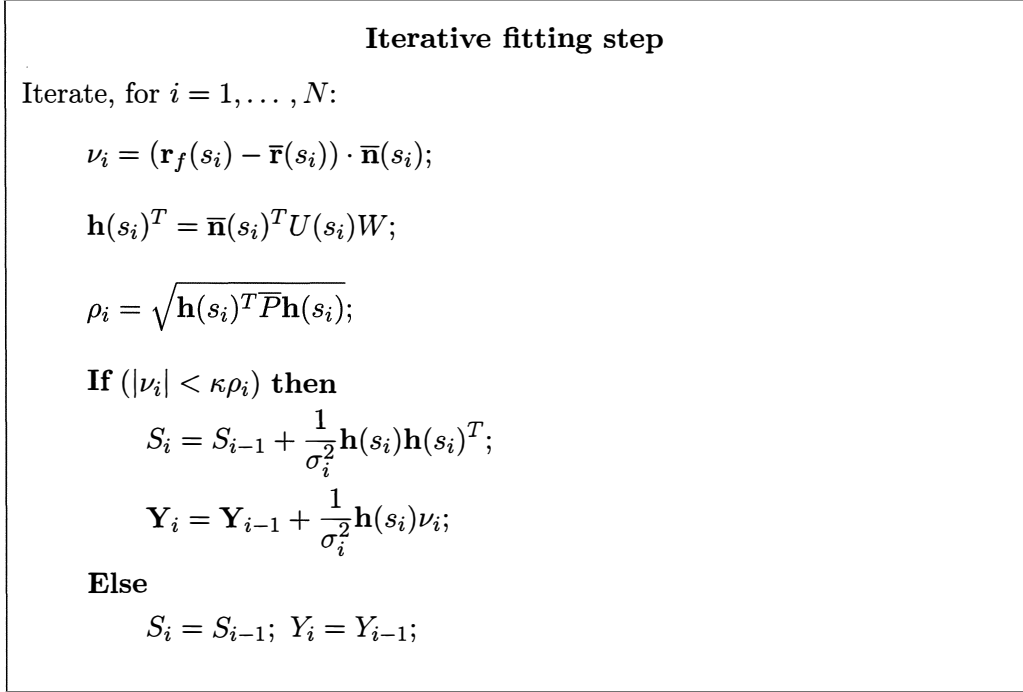


Figure 8.6: Curve fitting with outliers. *The recursive curve-fitting algorithm in figure 6.7 on page 127 can be modified to ignore outliers. Its iterative step is replaced by the one shown here. The search-line length factor is taken to be $\kappa = 2$ by default. (Note that $\bar{\mathbf{P}} = \bar{\mathbf{S}}^{-1}$ needs to have been computed at the start of the algorithm, and that $\bar{\mathbf{S}}$ must therefore be of full rank.)*

8.5 Learning the prior

The importance of a prior distribution is that it draws interpretations of data towards the more plausible shapes in a class of curves. The value of such stabilising behaviour is only as good as the prior model itself, embodied by the coefficients in the information matrix $\bar{\mathbf{S}}$. So far, we have proposed priors based on uniform, isotropic error ($\bar{\mathbf{S}} \propto \mathcal{H}$) and a modification that allows for an invariant subspace. These are reasonable choices if little is known precisely about the class of curves. If, however, a more specific prior could be obtained from actual snapshots of a curve in a series of representative configurations, stabilisation should be very much more effective, and this does indeed

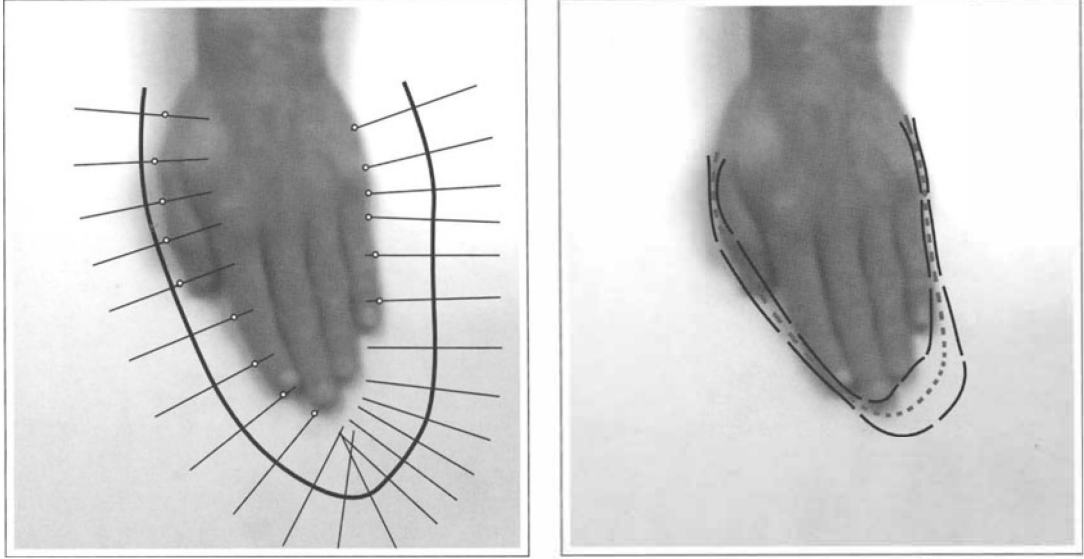


Figure 8.7: Validation failure causes a bulge in the posterior confidence region. Validation fails around the fingertips (left) resulting in a bulge in the 95% confidence region (right) such that it still contains the true edge of the hand.

prove to be the case.

The first step towards learning is to acquire a training set, a set of curves $\{\mathbf{X}_k, k = 1, \dots, M\}$, where $\mathbf{X}_k$ are vectors in a shape-space $\mathcal{S} = \mathcal{L}(W, \mathbf{Q}_0)$ which may either be the spline space $\mathcal{S}_Q$ or a subspace of $\mathcal{S}_Q$. The set is supposed to capture the outline being modelled in many characteristic poses. For current purposes, the order in which poses appear in the sequence is immaterial, though order will matter in later chapters when object dynamics are being modelled. One could think of each $\mathbf{X}_k$ as a contour drawn by hand on a given image but in practice the temporal tracking technology developed later allows these outlines to be acquired automatically.

If the prior distribution is assumed to be Gaussian $\mathcal{N}(\bar{\mathbf{X}}, \bar{P})$, then “maximum likelihood” estimators for the parameters of the distribution are given by the sample mean

$$\bar{\mathbf{X}} = \frac{1}{M} \sum_{k=1}^M \mathbf{X}_k \quad (8.22)$$

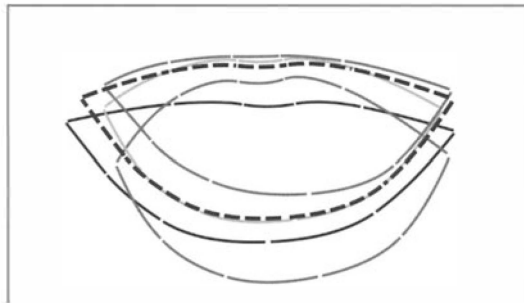


Figure 8.8: Sampling from a prior for lip shape Lips are tracked during speech, as in figure 1.10 on page 14, to capture a 60 second sequence comprising 3000 successive lip shapes. The sequence is used to estimate a Gaussian prior distribution in shape-space for lip shape; random sampling from the prior generates plausible lip configurations, as shown. (Data courtesy of Robert Kaucic.)

and sample covariance

$$\bar{P} = \frac{1}{M} \sum_{k=1}^M (\mathbf{X}_k - \bar{\mathbf{X}})(\mathbf{X}_k - \bar{\mathbf{X}})^T. \quad (8.23)$$

For example, in figure 8.8 a Gaussian prior distribution has been learned from a long sequence of lip motion during speech. The resulting prior is simulated by random sampling and the sampled curves are plausible lip configurations.

The matrix $\bar{P}$ is a $N_X \times N_X$ matrix and should be invertible so that $\bar{S} = \bar{P}^{-1}$ can be calculated as required for the recursive curve-fitting algorithm. However, from (8.23),

$$\text{rank}(\bar{P}) \leq M - 1$$

so that $\bar{P}$ certainly cannot be inverted unless $M > N_X$, and in practice one would expect a data set several times that minimum size. Alternatively, if a sufficiently large data set is not available and the estimated $\bar{P}$ is singular then one approach is to restrict the shape-space sufficiently that the reduced covariance matrix is no longer singular, and this is discussed next.

8.6 Principal Components Analysis (PCA)

Even if the estimated prior covariance matrix $\bar{P}$ is not technically singular, it is frequently close to singularity even when the training sequence is long ($M > N_X$). This

happens when the typical motions of the object under study are largely accounted for by a few independent modes of motion. If $\bar{P}$ is found to have a large condition number, larger say than 100, this suggests that, on the basis of the available data, the shape-space $\mathcal{S}$ in which data was collected is unnecessarily large. Efficiency of the matching algorithms described earlier, and of later tracking algorithms, would be considerably enhanced if it could be replaced by a smaller shape-space. The covariance matrix $\bar{P}$ can be used to determine a smaller shape-space $\mathcal{S}' = \mathcal{L}(W', \mathbf{Q}'_0) \subset \mathcal{S}$, a subspace of $\mathcal{S}$ with dimension N'_X , that spans, at least approximately, all of the shapes in a training sequence $\mathbf{X}_1, \dots, \mathbf{X}_M$. This idea was first introduced by Cootes and Taylor, in the special case of polygonal contour models, which they dubbed the “Point Distribution Model” or PDM. The PCA method is explained starting with “classical PCA” which is well-known and easily accessible in textbooks, via two refinements, to “ L_2 -norm PCA in shape-space,” the recommended method for dealing with curves. In the special case that $M \leq N_X$ and the covariance matrix $\bar{P}$ is actually singular, the simplest way to build a reduced shape-space is to take appropriate linear combinations of the frames in the training sequence — so-called key-frames, as described in chapter 4.

Classical PCA

The classical approach to PCA would allow the following version of the approximation problem to be solved. Given a long ($M > N_X$) training sequence, solve:

$$\min_{W', \mathbf{Q}'_0, \mathbf{X}'_1, \dots, \mathbf{X}'_{N'_X}} \left(\sum_{k=1}^M |\mathbf{Q}_k - \mathbf{Q}'_k|^2 \right) \quad (8.24)$$

where

$$\mathbf{Q}'_k = W' \mathbf{X}'_k + \mathbf{Q}'_0 \quad \text{and} \quad \mathbf{Q}_k = W \mathbf{X}_k + \mathbf{Q}_0.$$

The distance measure $|\cdot|$ is the Euclidean norm, that is, $|\mathbf{Q}|^2 \equiv \mathbf{Q}^T \mathbf{Q}$. The well known solution to this problem gives $\mathbf{Q}'_0 = \bar{\mathbf{Q}}$, the mean of the training sequence, and W' is a matrix whose columns are the first N'_X of the orthonormal eigenvectors of the covariance matrix

$$\Sigma = \frac{1}{M} \sum_{k=1}^M (\mathbf{Q}_k - \bar{\mathbf{Q}})(\mathbf{Q}_k - \bar{\mathbf{Q}})^T,$$

ordering the eigenvectors in descending order of their (necessarily positive) eigenvalues. The intuitive interpretation is that the eigenvalues represent variance in the training

set in the mutually orthogonal directions of the eigenvectors; the first N'_X eigenvectors form a basis for the subspace of dimension N'_X that “explains” as much as possible of the variance in the training set. Note that since the data $\mathbf{Q}_k$ all live within a shape-space of dimension N_X there will be at most N_X non-zero eigenvalues of Σ .

L_2 -norm PCA

In previous chapters, it has been argued that the induced L_2 -norm is the natural norm over spline space. In that case, the classical problem above is not entirely applicable. Instead, the approximation problem should be posed as follows:

$$\min_{W', \mathbf{Q}'_0, \mathbf{X}'_1, \dots, \mathbf{X}'_{N'_X}} \left(\sum_{k=1}^M \|\mathbf{Q}_k - \mathbf{Q}'_k\|^2 \right), \quad (8.25)$$

in which the L_2 -norm $\|\cdot\|$ has been substituted where Euclidean norm $|\cdot|$ appeared before. The solution to this problem, which can be derived from the solution to the classical problem (see below), is that $\mathbf{Q}'_0 = \overline{\mathbf{Q}}$ as before and W' is a matrix whose columns are the first N'_X eigenvectors of the matrix $\Sigma\mathcal{U}$.

L_2 -norm PCA over shape-space

Finally L_2 -norm PCA can be re-expressed in terms of training-set covariance $\overline{P}$ in shape-space, without recourse to the (considerably larger) covariance matrix Σ in spline space. This gives the recommended algorithm in figure 8.9. Results from the application of the algorithm to the lip-motion sequence are shown in figure 8.10. PCA analysis for full facial expression is illustrated in figure 8.11. In neither case are individual PCA components recognisable as particular expressions; rather they are mixtures of expressions. It is when they are taken as a set that they are meaningful, as a basis for the repertoire of commonly occurring deformations.

Remember that Euclidean norm $|\mathbf{X}|$ in shape-space has no clear geometrical meaning. As a result classical PCA applied to the shape-space training set $\mathbf{X}_1, \dots, \mathbf{X}_M$ would certainly not be meaningful.

Residual PCA

A constructive description of shape-space, in which the shape-matrix is composed from transformations of a template or from key-frames, is rather desirable because

Problem: given training data $\mathbf{X}_1, \dots, \mathbf{X}_M$ over shape-space $\mathcal{S}$, find a subspace $\mathcal{S}' = \mathcal{L}(W', \mathbf{Q}'_0)$ of dimension N'_X to minimise

$$\sum_{k=1}^M \|\mathbf{Q}_k - \mathbf{Q}'_k\|^2$$

where

$$\mathbf{Q}_k = W\mathbf{X}_k + \mathbf{Q}_0 \quad \text{and} \quad \mathbf{Q}'_k = W'\mathbf{X}'_k + \mathbf{Q}'_0.$$

Algorithm

1. Construct the training-set mean

$$\bar{\mathbf{X}} = \frac{1}{M} \sum_{k=1}^M \mathbf{X}_k.$$

2. Construct the training-set covariance

$$\bar{P} = \frac{1}{M} \sum_{k=1}^M (\mathbf{X}_k - \bar{\mathbf{X}})(\mathbf{X}_k - \bar{\mathbf{X}})^T.$$

3. Find eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_{N_X}$ of $\bar{P}\mathcal{H}$, in descending order of eigenvalue.

4. Construct $W'' = (\mathbf{v}_1, \dots, \mathbf{v}_{N'_X})$.

5. The parameters $\mathbf{Q}'_0$ and W' of the shape-subspace are then:

$$\begin{aligned} \mathbf{Q}'_0 &= W\bar{\mathbf{X}} + \mathbf{Q}_0 \\ W' &= WW''. \end{aligned}$$

Figure 8.9: Algorithm for L_2 PCA in shape-space.

each of the components of the shape-vector $\mathbf{X}$ has a clear interpretation. For example, the shape-space (4.20) composed of key-frames under Euclidean similarity is a constructive shape-space for which a given $\mathbf{X}$ represents an explicit combination of rigid transformation and facial expression. In contrast, deriving a shape-matrix from

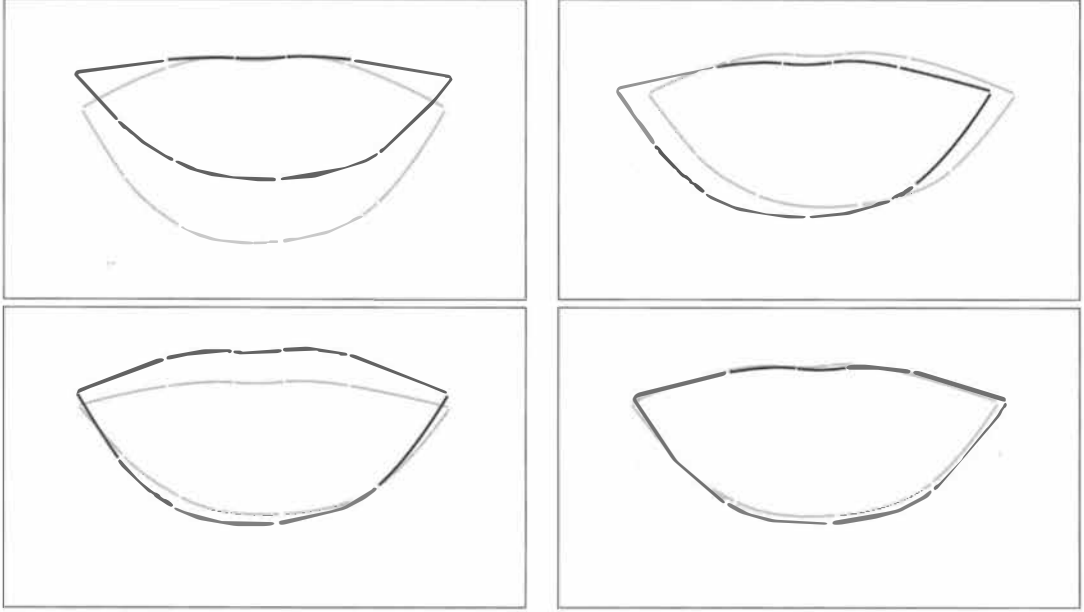


Figure 8.10: The lip-motion sequence of figure 8.8 on page 176 is used here in PCA. The first four eigenvectors, illustrated here, capture over 95% of the variance in the data set. (Eigenvectors are displayed here as displacements either side of the mean, with magnitude equal to their standard deviation over the sequence.) (Figures by courtesy of Robert Kaucic.)

PCA loses the clear interpretation, but has the advantage that no prior insight into likely transformations or deformations is required. It is possible, to some extent, to enjoy the best of both worlds. Residual PCA operates on a constructive shape-space that does not totally cover a certain data set, and fills in missing components by PCA. Then the constructive subspace retains its interpretation and only the residual components, covered by PCA, cannot be directly interpreted.

Given training data $\mathbf{X}_1, \dots, \mathbf{X}_M$, collected from a shape-space $\mathcal{S}$, and a constructive subspace $\mathcal{S}_c = \mathcal{L}(W^c, \mathbf{Q}_0^c)$, it is desired to augment $\mathcal{S}_c$, increasing its dimension by N'_X , to cover the training-data set more completely. Residual PCA achieves this by computing, after steps 1,2 of the algorithm of figure 8.9, the mean and variance of the residue of the training set:

$$\bar{\mathbf{X}}^r = (I - E^c)\bar{\mathbf{X}} \quad (8.26)$$

$$\bar{P}^r = (I - E^c)\bar{P}(I - E^c)^T \quad (8.27)$$

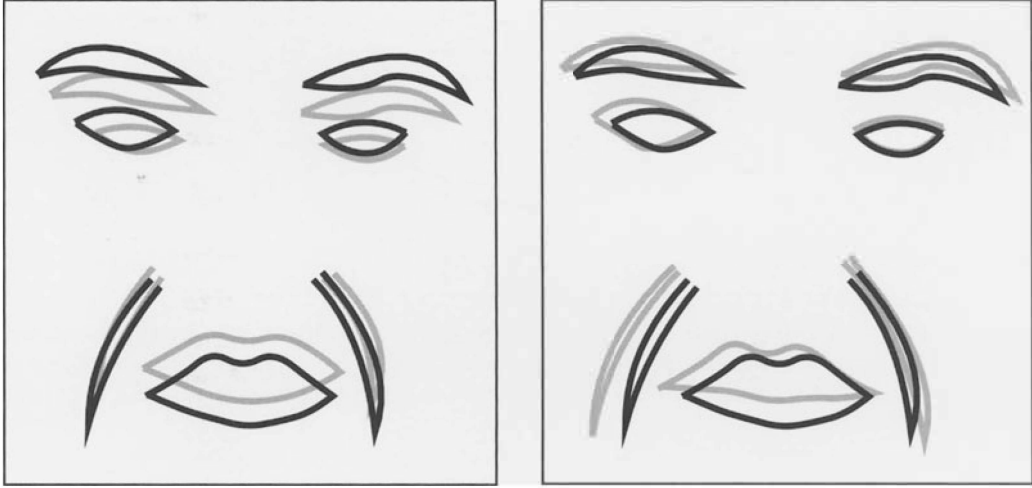


Figure 8.11: Facial expression as in figure 1.2 on page 6 is analysed by PCA. The first two eigenvectors are illustrated here. (Figures by courtesy of Benedicte Bascle.)

where $E^c = W(W^c)^+$ is the matrix for projection onto the subspace $\mathcal{S}_c$. Step 3 of the algorithm computes the first N'_X eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_{N'_X}$ now of $\overline{P}^r$, rather than $\overline{P}$ as in the original algorithm. Steps 4 and 5 proceed as before, computing the template $\mathbf{Q}'_0$ and shape-matrix W' which now form a shape-space for the *residual* variation in the training set that was not covered by $\mathcal{S}_c$. The augmented shape-space is then

$$\mathcal{S}_a = \mathcal{L}(W^a, \mathbf{Q}_0^a)$$

where

$$\mathbf{Q}_0^a = E^c \mathbf{Q}_0^c + W' \overline{\mathbf{X}}^r$$

and the shape-matrix

$$W^a = (W^c | W^r),$$

a concatenation of columns from the shape-matrices for the constructive and residual spaces.

As an illustration, the Residual PCA algorithm is applied to the lip-motion sequence used earlier. Instead of applying PCA directly to the motion sequence as in figure 8.13, a constructive shape-space $\mathcal{S}_c$, a two-dimensional space of rigid translations, is used. Translation accounts for about 58% of the motion of the training set

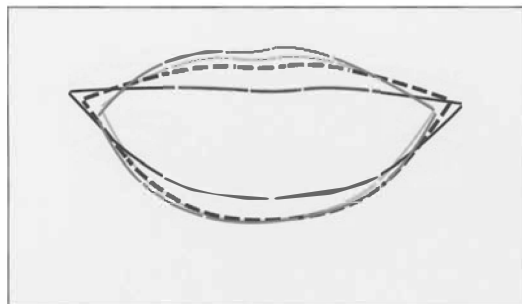


Figure 8.12: Sampling from a prior for lip shape, excluding translation *Random sampling illustrates how a learned prior generates plausible lip configurations, as in figure 8.8, but now with the rigid translation due to head motion excluded, leaving just the lip motions associated with speech.*

(horizontal 24%, vertical 34%) and since head motion is likely to be somewhat independent of speech, it is natural to try to discount it. The covariance $\overline{P}^r$ of the residue of the sequence with translation removed can be used in its own right to construct a prior, as in figure 8.13. Then the first two components from residual PCA, illustrated in the figure, can be expected to bear the bulk of the visual information that is related to speech.

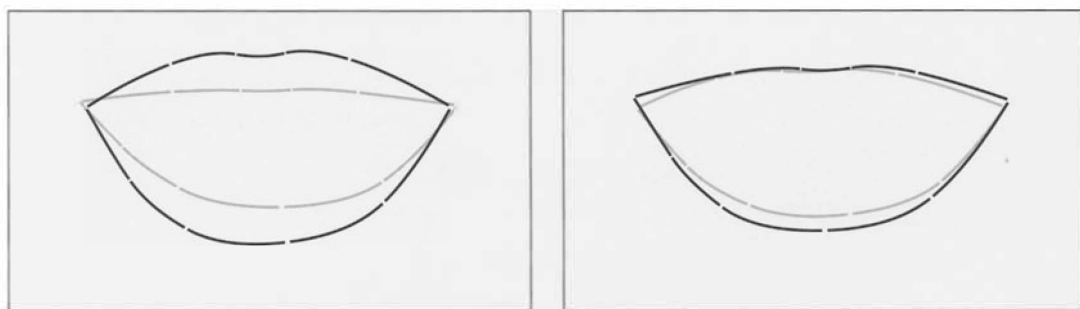


Figure 8.13: The lip-motion sequence is analysed by residual PCA. *All rigid translation is first excluded from the motion sequence by projection. Translation and the further 2 eigenvectors shown here account for over 95% of the variance in the data set.*

Derivation of PCA algorithms

First L_2 -norm PCA is derived from classical PCA. Noting that $\|\mathbf{Q}\| = |\mathcal{U}^{1/2}\mathbf{Q}|$, the problem (8.25) can be recast as a minimisation, in the form of the classical problem (8.24), of

$$\sum_{k=1}^M |\mathcal{U}^{1/2}\mathbf{Q}_k - \mathcal{U}^{1/2}\mathbf{Q}'_k|^2$$

with

$$\mathcal{U}^{1/2}\mathbf{Q}'_k = \mathcal{U}^{1/2}W'\mathbf{X}'_k + \mathcal{U}^{1/2}\mathbf{Q}'_0,$$

Its solution is therefore that

$$\mathcal{U}^{1/2}\mathbf{Q}'_0 = \mathcal{U}^{1/2}\overline{\mathbf{Q}}$$

and that $\mathcal{U}^{1/2}W'$ is made up of columns which are eigenvectors of the sample variance

$$\text{Var}(\mathcal{U}^{1/2}\mathbf{Q}_k) = \mathcal{U}^{1/2}\Sigma\mathcal{U}^{1/2}.$$

This means that $\mathbf{Q}'_0 = \overline{\mathbf{Q}}$. Furthermore, a column $\mathcal{U}^{1/2}w$ of $\mathcal{U}^{1/2}W'$ satisfies

$$\mathcal{U}^{1/2}\Sigma\mathcal{U}^{1/2}(\mathcal{U}^{1/2}w) = \lambda(\mathcal{U}^{1/2}w)$$

which simplifies to

$$\Sigma\mathcal{U}w = \lambda w,$$

each column of W' is an eigenvector of $\Sigma\mathcal{U}$, as claimed. Note that equivalently w is a *generalised* eigenvector of the real, symmetric matrices $\mathcal{U}\Sigma\mathcal{U}$ and $\mathcal{U}$ (i.e. solutions of $\mathcal{U}\Sigma\mathcal{U}\mathbf{v} = \lambda\mathcal{U}\mathbf{v}$) and hence the eigenvalues are still positive, so that their descending order remains well-defined.

The shape-space version of the algorithm (figure 8.9) is obtained from the L_2 -norm algorithm above by substituting

$$\mathbf{Q}_k = W\mathbf{X}_k + \mathbf{Q}_0 \quad \text{and} \quad \mathbf{Q}'_k = W'\mathbf{X}'_k + \mathbf{Q}'_0$$

so that

$$\overline{\mathbf{Q}} \rightarrow W\overline{\mathbf{X}} + \mathbf{Q}_0,$$

as in the algorithm, and

$$\Sigma\mathcal{U} \rightarrow W\overline{P}W^T\mathcal{U}$$

whose eigenvectors give W' . Writing $W' = WW''$, W'' must therefore be composed of eigenvectors of $\overline{P}W^T\mathcal{U}W$, but $W^T\mathcal{U}W = \mathcal{H}$ and this gives the required form for W' .



Bibliographic notes

The chapter is based on probabilistic analysis — for a review of basic probability, there is an excellent introductory book (Papoulis, 1990).

The use of a statistically estimated $\bar{P}$ in a Gaussian prior is equivalent to replacing the norm $\|\cdot\|$ in the regulariser of chapter 6 with a “Mahalanobis metric” (Rao, 1973) that emphasises improbable distortions, so that they are penalised more than probable ones.

Confidence regions for shapes were delimited by parallel curves. These are a well-known geometrical construct; one pitfall is that a smooth base curve can have offset curves that are not smooth but contain cusps (Bruce and Giblin, 1984). They are also known in CAD as “offset” curves (Faux and Pratt, 1979).

Principal Components Analysis (PCA) is a standard statistical technique (Rao, 1973). The idea of constructing a Gaussian curve model of reduced dimension, using PCA, was first developed for the case of polygonal outlines and termed the “Point Distribution Model” or PDM (Cootes et al., 1993), and subsequently extended to splines (Baumberg and Hogg, 1994). A related idea based on image intensities rather than curves is the eigen-image (Turk and Pentland, 1991) which can be used directly in tracking e.g. (Bregler and Omohundro, 1994; Black and Jepson, 1996). PCA is sometimes applied jointly to outline shape and image intensity distributions (Lanitis et al., 1995; Beymer and Poggio, 1995; Vetter and Poggio, 1996), to impressive effect. In standard PCA, it is common to pre-process the data by applying a diagonal scaling transformation (Ripley, 1996). This proved to be too restrictive for building curve models, for which general linear transformations were needed. In particular the idea of performing PCA in the L_2 norm, using the metric matrix $\mathcal{H}$ was developed in (Baumberg and Hogg, 1995a), and this is equivalent to scaling the data by $\mathcal{H}^{1/2}$. Generalised eigenvalues (Golub and van Loan, 1989) are used to combine scaling and PCA efficiently.

Shape priors discussed here have had a Gaussian form in shape-space. In the case of a norm-squared density over spline space ($S \propto \mathcal{U}$), the prior is actually a Gaussian Markov Random Field (MRF), of second-order in the case of quadratic splines. MRFs have been used widely for modelling prior distributions for curves (Grenander et al., 1991; Ripley and Sutherland, 1990; Storvik, 1994).

Appendix A

Mathematical background

A.1 Vectors and matrices

A brief summary of vector and matrix conventions and operations is given here. An excellent handbook for vector and matrix computation is (Barnett, 1990) and readers should refer to it for details. An alternative that may also be found helpful is (Golub and van Loan, 1989). The vector and matrix operations described should be available in appropriate programming languages such as **matlab**.

Vectors Throughout the book, vectors are denoted in bold, for example

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{or} \quad \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Scalar product and vector product take their usual meanings and are denoted

$$\mathbf{a} \cdot \mathbf{b} \quad \text{and} \quad \mathbf{a} \times \mathbf{b}$$

respectively. In three dimensions, for instance,

$$\mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3.$$

and

$$\mathbf{a} \times \mathbf{b} = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_3)^T.$$

The magnitude of a vector may be measured via its **Euclidean norm**:

$$|\mathbf{r}| = \sqrt{\mathbf{r} \cdot \mathbf{r}}$$

and a vector $\mathbf{r}$ for which $|\mathbf{r}| = 1$ is said to be normalised, or a “unit” vector.

Matrices Matrices are generally non-bold capitals, for example A , with components denoted A_{ij} . The transpose A^T is defined by

$$A_{ij}^T = A_{ji}.$$

The **rank** of a matrix A is the number of linearly independent vectors that comprise its columns.

A matrix operation that is frequently useful is the **Kronecker product**

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B & \dots \\ A_{21}B & A_{22}B & \dots \\ \dots & \dots & \dots \end{pmatrix} \quad (\text{A.1})$$

which combines two arrays of dimension $M_1 \times N_1$ and $M_2 \times N_2$ to make a larger one of dimension $M_1 M_2 \times N_1 N_2$.

Linear equations The linear simultaneous equations

$$A\mathbf{x} = \mathbf{b}$$

have a unique solution when A is square and is non-singular — that is, $\det A \neq 0$, where $\det A$ is the **determinant** of A . Then a solution can be found using the standard inverse

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

If there is no solution, as may happen when A is not square, there may nonetheless be a unique, optimal, approximate solution which is expressed using a **pseudo-inverse** A^+ :

$$\mathbf{x} = A^+\mathbf{b} \text{ where } A^+ = (A^T A)^{-1} A^T.$$

(More general definitions of pseudo-inverse can be made, but are not used in this book.)

Rotation matrices Matrices for rotation about x , y and z axes are respectively denoted R_x , R_y and R_z where, for example,

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.2})$$

so that a point in three dimensions given by a vector $\mathbf{r} = (x, y, z)^T$ is transformed to a rotated point

$$\mathbf{r}' = R_z(\theta)\mathbf{r}.$$

In two dimensions, a rotation is a 2×2 matrix

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Rotation matrices have the property that they are **orthogonal**, satisfying $R^T R = I$. Generally, an orthogonal matrix U satisfies $U^T U = I$ and, in three dimensions, can be interpreted as a rotation (when $\det U = 1$) or a combination of rotation and reflection ($\det U = -1$).

Eigenvalues and eigenvectors The eigenvalues λ_n and eigenvectors $\mathbf{u}_n$ of a square $N \times N$ matrix A are defined as satisfying

$$A\mathbf{u}_n = \lambda_n \mathbf{u}_n, \quad n = 1, \dots, N.$$

Eigenvalues and eigenvectors may have complex values, unless A is **symmetric** ($A^T = A$) in which case they are guaranteed to be real.

The **trace** of the matrix is defined to be

$$\text{tr}(A) = \sum_{n=1}^N A_{nn}$$

and has the property that

$$\text{tr}(A) = \sum_{n=1}^N \lambda_n.$$

Diagonalisation Eigenvalues and eigenvectors can be used to decompose a square matrix A as

$$A = UDU^{-1}$$

where D is the “diagonal” matrix

$$D = \text{diag}(\lambda_1, \dots, \lambda_N)$$

with the eigenvalues along the diagonal and zeros elsewhere. The matrix U consists of columns which are normalised eigenvectors of A . One important application of the diagonal form is in computing powers of A :

$$A^p = UD^pU^{-1} \quad \text{where} \quad D^p = \text{diag}(\lambda_1^p, \dots, \lambda_N^p).$$

Setting $p = \frac{1}{2}$ allows a **square root** of A to be computed.

Singular value decomposition (SVD) An alternative form of decomposition of a matrix A is the SVD, which applies not only to square matrices but also to rectangular ones of size $M \times N$. It has the form

$$A = UDV$$

where U is an $M \times M$ matrix, D is a diagonal $M \times N$ matrix and V is an $N \times N$ matrix. Both U and V are orthogonal matrices. The diagonal values of D are $D_{nn} = \sigma_n$ where $\lambda_n = \sigma_n^2$ are eigenvalues of the symmetric matrix $A^T A$ and are hence guaranteed to be positive.

A measure of the “size” of A is its **spectral radius** $\|A\|_2 = \sqrt{\lambda_1}$ where λ_1 is the largest eigenvalue of $A^T A$. The condition for an iterative process involving A , in which arbitrarily large powers A^n of A are applied to vectors, to be stable is that $\|A\|_2 < 1$.

A.2 B-spline basis functions

Useful introductory reference books on splines are (Faux and Pratt, 1979; Foley et al., 1990). An excellent, comprehensive reference is (Bartels et al., 1987).

In chapter 3, spline functions are written as a linear combination of a number of spline “basis functions.” Basis functions are constructed using the following general rule which can be used to define any arbitrary set of polynomial splines. Let $B_{n,d}$ be the n th basis function for a spline of order d . Then for a spline with single knots of unit spacing, the following recursive rule applies:

Ground instance

$$B_{n,1}(s) = \begin{cases} 1 & \text{if } n \leq s < n+1 \\ 0 & \text{otherwise} \end{cases}$$

Inductive step

$$B_{n,d}(s) = \frac{(s-n)B_{n,d-1}(s) + (n+d-s)B_{n+1,d-1}(s)}{d-1}$$

and some examples are shown in figure A.1. These functions satisfy the following conditions:

Support $B_{n,d}(s) = 0$ for $s \notin [n, n+d)$

Positivity $B_{n,d}(s) \geq 0$ for all s

Normalisation $\sum_{-\infty}^{\infty} B_{n,d}(s) = 1$ for all s

Translational invariance $B_{n+1,d}(s) = B_{n,d}(s-1)$ for all s

and further, there is a smoothness constraint for $d > 1$, namely that $B_{n,d}$ has continuous $(d-2)$ th derivative for all s and all $d > 1$.

Non-uniform B-spline functions

The spline basis functions generated above, for which the knots are uniformly spaced at unit intervals, can be generalised to produce spline functions with arbitrary knot spacing. Consider a spline with N_K knots at positions $k_0 \leq k_1 \leq \dots \leq k_{N_K-1}$, then the recursive rule becomes

Ground instance

$$B_{n,1}(s) = \begin{cases} 1 & \text{if } k_n \leq s < k_{n+1} \\ 0 & \text{otherwise} \end{cases}$$

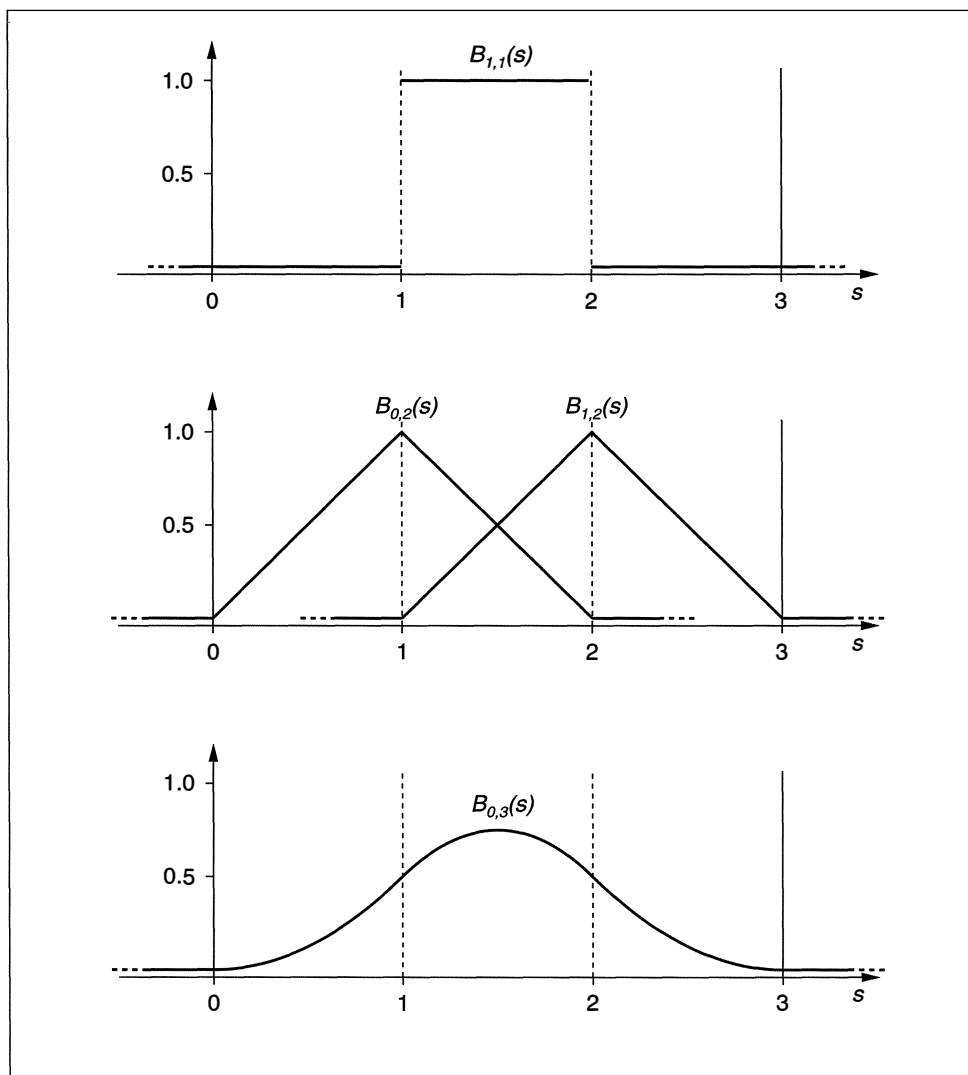


Figure A.1: A spline basis function $B_{n,d}$ of order d is built up recursively from basis functions of lower order.

Inductive step

$$B_{n,d}(s) = \frac{(s - k_n)B_{n,d-1}(s)}{k_{n+d-1} - k_n} + \frac{(k_{n+d} - s)B_{n+1,d-1}(s)}{k_{n+d} - k_{n+1}}$$

which reduces to the uniform case when $k_n = n$. The rule can be used to generate spline functions with knots of multiplicity m by setting m consecutive k_n to be equal. (Terms in the inductive step are zero when the denominator is zero. The validity of this can be shown by taking a limit as the knots approach one another — it is easy to see by induction that the basis function in the numerator is identically zero whenever the denominator is zero). The conditions of positivity and normalisation still hold in general, and now the support of the basis function $B_{n,d}$ is $[k_n, k_{n+d})$. The basis functions are clearly no longer necessarily translated copies of each other, however, and the introduction of a multiple knot reduces the smoothness of a basis function; the function is C^{d-1-m} at a knot of multiplicity m . This weakening of the smoothness property is the motivation for using multiple knots; it permits B-spline functions, and therefore curves, with sharp corners and discontinuities.

An implementation of B-spline functions

The recursive rule for generating $B_{n,d}$ can be converted into an algorithm by expressing each basis function as a sequence of polynomials $p_n(s)$ defined over the intervals $[k_n, k_{n+1})$. Since the support¹ of $B_{n,d}$ is $[k_n, k_{n+d})$, any spline basis function of order d can be represented using just d polynomials $B_{n,d}^\sigma$, one for each of the d spans S_σ in the support of $B_{n,d}$. Now the inductive step of the rule can be applied, over each interval in turn, to obtain each of the $B_{n,d}^\sigma$.

Where a B-spline contains multiple knots, some of the inter-knot intervals have zero length, so it is convenient to introduce the concept of “spans”. These correspond to the non-empty inter-knot intervals above, and the span ends are called “breakpoints”. A B-spline function, therefore, is a piecewise polynomial curve made up of a series of L spans $S_0 \dots S_{L-1}$ connected at breakpoints $s_0 < s_1 < \dots < s_L$. We adopt the convention that all spans are unit length, ($s_i = i$), so the basis functions making up a spline are uniquely determined by the knot multiplicities $m_0 \dots m_L$ at the breakpoints. A periodic B-spline function is constructed by considering the basis functions to be

¹The support of a function is the interval over which it is non-zero.

periodic over the interval $[0, L]$. A periodic B-spline function must have $m_0 = m_L$ and it has “multiple knot count”

$$M = \sum_1^L (m_i - 1)$$

while for an aperiodic spline $m_0 = m_L = d$ to control the boundary conditions of the spline, and

$$M = \sum_0^L (m_i - 1).$$

An L span B-spline is a linear combination of N_B basis functions, where

$$N_B = L + M = N_K - m_0$$

for a periodic spline, and

$$N_B = L + M + 1 - d = N_K - d$$

for an aperiodic spline. Thus, for example, a simple L span aperiodic quadratic B-spline is a linear combination of $N_B = L + 2$ basis functions (see figure 3.6 on page 48). The relationship between spans, knots and basis functions is illustrated for two cases in figures A.2 and A.3.

Building a spline function from basis functions

B-spline functions can be evaluated efficiently using “span matrices.” Over the span S_σ , any spline function is a linear combination of the basis functions $B_{b_\sigma, d} \dots B_{b_\sigma + d - 1, d}$ where

$$b_\sigma = \left(\sum_{i=0}^{\sigma} m_i \right) - d$$

so

$$x(s)^{[\sigma, \sigma+1)} = x(s)^\sigma = \sum_{b_\sigma}^{b_\sigma + d - 1} x_i B_{i, d}(s)$$

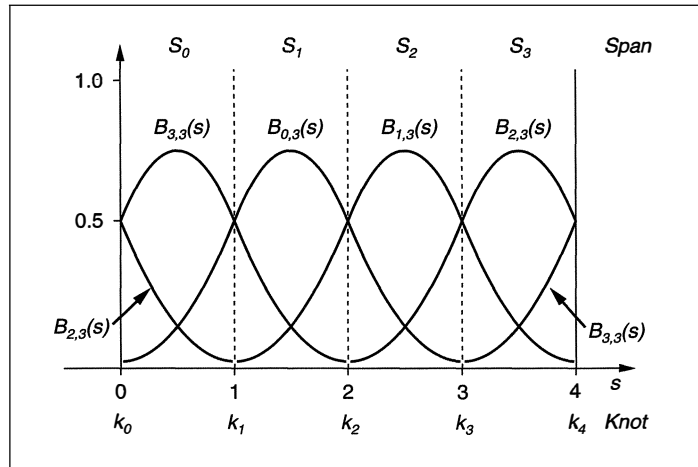


Figure A.2: A simple periodic B-spline with no multiple knots has $L = 4$ spans, $N_K = 5$ knots, and is a combination of $N_B = 4$ (periodic) basis functions.

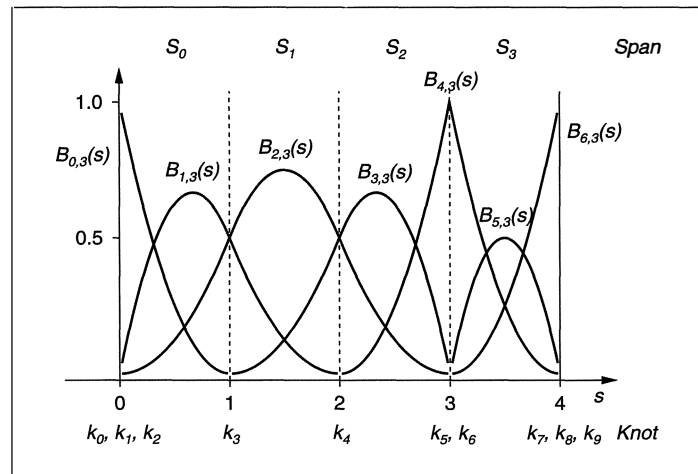


Figure A.3: An aperiodic spline must have knots of multiplicity d at its endpoints. Here there is also a double knot between the third and fourth spans, leading to a discontinuity in the first derivative of the function. Here $L = 4$, $N_K = 10$, $N_B = 7$.

(with obvious variations for periodic splines). For each span, therefore, we can compute a $d \times d$ span matrix B_σ^S such that

$$x(s + \sigma)^\sigma = (1 \ s \ \dots \ s^{d-1}) B_\sigma^S \begin{pmatrix} x_{b_\sigma} \\ x_{b_\sigma+1} \\ \vdots \\ x_{b_\sigma+d-1} \end{pmatrix}$$

where the i th column of the span matrix corresponds to the polynomial coefficients of the basis function $B_{b_\sigma+i-1,d}$ over the interval of that span (in practice it is convenient to define each span matrix over the interval $[0, 1)$). The span matrices for the spline in figure A.3 are as follows:

$$\begin{aligned} B_0^S &= \begin{pmatrix} 1.00 & 0.00 & 0.00 \\ -2.00 & 2.00 & 0.00 \\ 1.00 & -1.50 & 0.50 \end{pmatrix} \\ B_1^S &= \begin{pmatrix} 0.50 & 0.50 & 0.00 \\ -1.00 & 1.00 & 0.00 \\ 0.50 & -1.00 & 0.50 \end{pmatrix} \\ B_2^S &= \begin{pmatrix} 0.50 & 0.50 & 0.00 \\ -1.00 & 1.00 & 0.00 \\ 0.50 & -1.50 & 1.00 \end{pmatrix} \\ B_3^S &= \begin{pmatrix} 1.00 & 0.00 & 0.00 \\ -2.00 & 2.00 & 0.00 \\ 1.00 & -2.00 & 1.00 \end{pmatrix} \end{aligned}$$

and the algorithm used to calculate them is given in figure A.4. Once span matrices have been computed off-line, the spline can be evaluated efficiently at any values of s

To calculate span matrices for a non-periodic B-spline of order d with knot multiplicities m_i , $0 \leq i \leq L$.

1. **Calculate** the knot values k_i :

(a) **Initialise:** $p = 0$, $q = 0$

(b) **For** $i = 0 \dots L$

i. **For** $j = 1 \dots m_i$

$k_p = q$, $p = p + 1$

ii. $q = q + 1$

2. **For** each span $\sigma = 0 \dots L - 1$:

(a) **Find** the index b_σ of the first basis function whose support includes the span.

$$b_\sigma = \left(\sum_0^\sigma m_i \right) - d$$

(b) **For** $i = 1 \dots d$ recursively calculate the basis polynomial $B_{b_\sigma+i-1,d}^\sigma$ for span σ using the following rule

i. **Ground instance**

$$B_{n,1}^\sigma(s) = \begin{cases} 1 & \text{if } k_n \leq \sigma < k_{n+1} \\ 0 & \text{otherwise} \end{cases}$$

ii. **Recursive rule**

$$B_{n,d}^\sigma(s) = \frac{(s + \sigma - k_n)B_{n,d-1}^\sigma(s)}{k_{n+d-1} - k_n} + \frac{(k_{n+d} - s - \sigma)B_{n+1,d-1}^\sigma(s)}{k_{n+d} - k_{n+1}}$$

where terms are zero when the denominator is zero.

iii. **Store** the coefficients of $B_{b_\sigma+i-1,d}^\sigma$ as the i th column of the $d \times d$ span matrix B_σ^S , where the top row corresponds to the constant polynomial coefficient.

Figure A.4: Algorithm to calculate span matrices for aperiodic B-splines. *Obvious modifications must be made for the periodic case.*

and $x_0 \dots x_{N_B-1}$. For notational purposes it is convenient also to define the $d \times N_B$ “placement matrices” G_σ :

$$(G_\sigma)_{ij} = \begin{cases} 1 & \text{if } i - b_\sigma = j \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.3})$$

so that

$$x(s + \sigma) = (1 \ s \ \dots \ s^{d-1}) B_\sigma^S G_\sigma \mathbf{Q}$$

where $0 \leq s < 1$. The derivative of the function can be calculated as

$$x'(s + \sigma) = (0 \ 1 \ \dots \ (d-1)s^{d-2}) B_\sigma^S G_\sigma \mathbf{Q}$$

and so when considering a spline curve,

$$\begin{pmatrix} x \\ y \end{pmatrix} = (1 \ s \ \dots \ s^{d-1} \ 1 \ s \ \dots \ s^{d-1}) \begin{pmatrix} B_\sigma^S G_\sigma & 0 \\ 0 & B_\sigma^S G_\sigma \end{pmatrix} \begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix},$$

the tangent to the curve is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = (0 \ 1 \ \dots \ (d-1)s^{d-2} \ 0 \ 1 \ \dots \ (d-1)s^{d-2}) \begin{pmatrix} B_\sigma^S G_\sigma & 0 \\ 0 & B_\sigma^S G_\sigma \end{pmatrix} \begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix},$$

and the normal is given by

$$\begin{pmatrix} n_x \\ n_y \end{pmatrix} = \begin{pmatrix} -y' \\ x' \end{pmatrix}.$$

Calculating the spline metric matrix

Using the span matrices it is straightforward to compute the spline metric matrix $\mathcal{B}$, where

$$\mathcal{B} = \frac{1}{L} \int_0^L \mathbf{B}(s) \mathbf{B}(s)^T ds \quad (\text{A.4})$$

as described on page 50.

$$\begin{aligned}\mathcal{B} &= \frac{1}{L} \sum_{\sigma=0}^{L-1} \left(\int_0^1 \mathbf{B}(s+\sigma) \mathbf{B}(s+\sigma)^T ds \right) \\ &= \frac{1}{L} \sum_{\sigma=0}^{L-1} G_\sigma^T (B_\sigma^S)^T \mathcal{P} B_\sigma^S G_\sigma\end{aligned}$$

where

$$\mathcal{P} = \int_0^1 \begin{pmatrix} 1 \\ \vdots \\ s^{d-1} \end{pmatrix} \begin{pmatrix} 1 & \dots & s^{d-1} \end{pmatrix} ds,$$

the “Hilbert” matrix (Barnett, 1990) whose coefficients are

$$\mathcal{P}_{ij} = \frac{1}{i+j-1}.$$

Similarly, the matrix $\mathcal{B}'$ used on page 65 to define the area coefficients $\mathcal{A}$, was defined as

$$\mathcal{B}' = \frac{1}{L} \int_0^L \mathbf{B}(s) \mathbf{B}'^T(s) ds$$

and may be calculated as follows:

$$\mathcal{B} = \frac{1}{L} \sum_{\sigma=0}^{L-1} G_\sigma^T (B_\sigma^S)^T \mathcal{P}' B_\sigma^S G_\sigma$$

where

$$\begin{aligned}\mathcal{P}' &= \int_0^1 \begin{pmatrix} 1 \\ \vdots \\ s^{d-1} \end{pmatrix} \begin{pmatrix} 0 & \dots & (d-1)s^{d-2} \end{pmatrix} ds, \text{ so} \\ \mathcal{P}'_{ij} &= \begin{cases} 0 & \text{if } i = j = 1 \\ \frac{j-1}{i+j-2} & \text{otherwise} \end{cases}.\end{aligned}$$

A.3 Probability

An excellent introductory text on probability is (Papoulis, 1990). It is impossible to cover the necessary ground here, but since much of the argument in the book is probabilistic, a few basic concepts are reviewed here.

Probability distributions A continuous random variable x taking real values $x \in \mathbb{R}$ has a probability distribution defined by its **density function** $p(x) \geq 0$. Its interpretation is that, for an interval $\mathcal{I} = [a, b]$:

$$P(x \in \mathcal{I}) = \int_a^b p(x) dx.$$

This definition extends to a multi-dimensional random variable $\mathbf{X} \in \mathbb{R}^{N_x}$ so that, for a subset $\mathcal{I} \in \mathbb{R}^{N_x}$:

$$P(\mathbf{X} \in \mathcal{I}) = \int_{\mathcal{I}} p(\mathbf{X}) d\mathbf{X}.$$

Since $\mathbf{X}$ has to take some value, p must satisfy the normalisation property that

$$\int_{\mathbb{R}^{N_x}} p(\mathbf{X}) d\mathbf{X} = 1.$$

A **conditional** distribution for X specifies the probable values of $\mathbf{X}$ *given* that the value of some related variable $\mathbf{Y}$ is known and is defined by the density $p(\mathbf{X}|\mathbf{Y})$. This is interpreted, as before, via integration:

$$P(\mathbf{X} \in \mathcal{I}|\mathbf{Y}) = \int_{\mathcal{I}} p(\mathbf{X}|\mathbf{Y}) d\mathbf{X}.$$

The associated normalisation property is

$$\int_{\mathbb{R}^{N_x}} p(\mathbf{X}|\mathbf{Y}) d\mathbf{X} = 1.$$

Mean and variance The expectation or mean of the random variable $\mathbf{X}$, denoted $\mathcal{E}[\mathbf{X}]$, is

$$\mathcal{E}[\mathbf{X}] = \int_{\mathbb{R}^{N_x}} p(\mathbf{X}) d\mathbf{X}$$

which is a linear operation so that

$$\mathcal{E}[A\mathbf{X} + \mathbf{b}] = A\mathcal{E}[\mathbf{X}] + \mathbf{b}.$$

The variance of $\mathbf{X}$, denoted $\mathcal{V}[\mathbf{X}]$ is defined as an expectation:

$$\mathcal{V}[\mathbf{X}] = \mathcal{E}[(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T]$$

where $\bar{\mathbf{X}} = \mathcal{E}[\mathbf{X}]$. It scales quadratically, as $\mathcal{V}[A\mathbf{X} + \mathbf{b}] = A\mathcal{V}[\mathbf{X}]A^T$, and is invariant to the additive constant $\mathbf{b}$, naturally enough since it is a measure of “spread” about the mean. It is also known as the “covariance matrix” of $\mathbf{X}$ and must be symmetric and “positive semi-definite” (all eigenvalues positive or zero).

Bayes’ rule Suppose a density $p(\mathbf{X})$ is given, based on prior knowledge of the state $\mathbf{X}$ of some system and its likely values. Then suppose that observations $\mathbf{Z}$ are made from an imperfect sensing device which is characterised by its **observation density** $p(\mathbf{Z}|\mathbf{X})$, specifying the likely range of observations *given* a particular system state $\mathbf{X}$. Then Bayes’ rule gives the **posterior density** $p(\mathbf{X}|\mathbf{Z})$:

$$p(\mathbf{X}|\mathbf{Z}) = kp(\mathbf{Z}|\mathbf{X})p(\mathbf{X}),$$

where k is a constant, not dependent on $\mathbf{X}$, whose value can be determined if need be by insisting that the posterior be normalised. Note that $p(\mathbf{Z}|\mathbf{X})$ is also known as a **likelihood** function for $\mathbf{X}$.

Estimation When both the prior and observation density are available, a common way to estimate the value of $\mathbf{X}$ given observations $\mathbf{Z}$ is simply to find the $\mathbf{X}$ that maximises the posterior:

$$\hat{\mathbf{X}} = \arg \max_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}).$$

This is known as the MAP (Maximum A Posteriori) estimate. Alternatively, if no prior is available, an estimator can be defined by

$$\hat{\mathbf{X}} = \arg \max_{\mathbf{X}} p(\mathbf{Z}|\mathbf{X}),$$

the MLE (Maximum Likelihood Estimator).

Normal distribution Much use is made in the book of multi-variate normal or Gaussian distributions. A vector variable $\mathbf{X}$ distributed as a Gaussian is denoted

$$\mathbf{X} \sim \mathcal{N}(\bar{\mathbf{X}}, P) \tag{A.5}$$

where $\bar{\mathbf{X}}$ is the mean of the distribution and P is its covariance matrix, assumed non-singular. The density function for $\mathbf{X}$ is

$$p(\mathbf{X}) = \frac{1}{\sqrt{2\pi}^{N_X}} \frac{1}{\sqrt{\det P}} \exp -\frac{1}{2}(\mathbf{X} - \bar{\mathbf{X}})^T S(\mathbf{X} - \bar{\mathbf{X}})$$

where $S = P^{-1}$, the **information matrix**.

Alternatively, given a vector $\mathbf{w}$ of N_X independent standard normal distributions, so that each $w_n \sim \mathcal{N}(0, 1)$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, I_{N_X})$, $\mathbf{X}$ can be described as a linear transformation of $\mathbf{w}$:

$$\mathbf{X} = B\mathbf{w} + \bar{\mathbf{X}}$$

where $B = \sqrt{P}$. Circles $|\mathbf{w}| < c$ map to **confidence** ellipsoids in $\mathbf{X}$ space, regions which contain the value of $\mathbf{X}$ with probability $\chi_{N_X}^2(c)$, where χ_ν^2 is the “chi-squared distribution function” for ν degrees of freedom, and can be found in statistical tables. For example, for $N_X = 2$,

$$P(|\mathbf{w}| < 2) = 86\% \quad \text{and} \quad P(|\mathbf{w}| < 3) = 99\%.$$

Appendix B

Stochastic dynamical systems

B.1 Continuous-time first-order dynamics

A first-order AR process (9.7) can be regarded as a first-order “stochastic differential equation” (SDE) in continuous time that has been sampled at regular intervals. If the sampling interval is τ so that $t_k = k\tau$, then the AR process is obtained by integrating the SDE over successive sampling intervals. The SDE is expressed as

$$\dot{\mathbf{X}} = F(\mathbf{X} - \bar{\mathbf{X}}) + G\dot{\mathbf{w}} \quad (\text{B.1})$$

where $\mathbf{X}(t)$ is a vector in shape-space, F and G are $N_X \times N_X$ matrices and $\mathbf{w}(t)$ is a N_X -dimensional vector of independent, univariate Brownian processes in continuous time. A univariate Brownian process w has the property that the value $w(t)$ has a Gaussian distribution with $\mathcal{E}[w(t)] = 0$ and $\mathcal{V}[w(t)] = t$. The derivative $\dot{w}(t)$ is a “white noise” signal, that is one with equal power at all frequencies. The coefficients F are the deterministic parameters of the process, in the sense that its eigenvalues λ_i are the so-called “poles” of the AR process, constants with units of inverse time that represent the rates of decay of the various characteristic motions of the system. (This applies to the case that all poles are real-valued and negative. Any real positive pole will cause the process to be unstable. There is also the possibility of complex poles, representing oscillations or damped oscillations.) The matrix G represents a coupling to the multi-dimensional white noise $\dot{\mathbf{w}}$ that is driving the dynamical system. As in the discrete case, there is a mean-state and a Riccati equation for continuous time:

$$\dot{\hat{\mathbf{X}}} = F(\hat{\mathbf{X}} - \bar{\mathbf{X}}) \quad \text{and} \quad \dot{P} = FP + PF^T + Q \quad (\text{B.2})$$

where the “covariance coefficient” $Q = GG^T$.

Conversion between continuous and discrete time

A continuous-time SDE can be converted to a discrete-time form (Gelb, 1974; Astrom and Wittenmark, 1984) by computing A and C directly from F and Q :

$$A = \exp F\tau \quad \text{and} \quad C = \int_0^\tau (\exp Ft) Q (\exp F^T t) dt. \quad (\text{B.3})$$

It is possible to evaluate the integral for C exactly by diagonalising F but in practice the following approximation, to lowest order in τ , is convenient (and particularly so for the second-order process):

$$A = (I - F\tau)^{-1} \quad \text{and} \quad C = Q\tau \quad \text{so that} \quad B = G\sqrt{\tau}. \quad (\text{B.4})$$

The approximation for A is known as the “backward difference” approximation and is preferable to the more obvious “forward difference” $A = F\tau$ because it preserves stability: that is, any SDE that is stable is approximated as a stable AR process, with $\|A\|_2 < 1$.

Power spectrum

In chapter 9, a form (9.12) on page 199 for the power spectrum of a first-order ARP is used. That form is derived briefly here. Restricting $\mathbf{X}(t)$ in the continuous process above to be one-dimensional $X(t)$, so that F and G are scalar coefficients, suppose that

$$X(t) \propto \exp 2\pi i f t,$$

and set its mean to zero for simplicity. Then (B.1) becomes

$$2\pi i f X = FX + G\dot{w}$$

so that

$$X = \frac{G\dot{w}}{2\pi i f - F}$$

and the power spectrum

$$S_{XX}(f) = \left| \frac{G}{2\pi i f - F} \right|^2 S_{\dot{w}\dot{w}}(f),$$

where $|\cdot|$ denotes complex modulus. Now the power spectrum $S_{\dot{w}\dot{w}}(f)$ of white noise is constant, so

$$S_{XX}(f) \propto \frac{G^2}{4\pi^2 f^2 + F^2}$$

which can be rewritten directly in the required form.

B.2 Second-order dynamics in continuous time

A second-order SDE can be written, as it was in the discrete case, as a first-order one in a suitable state-space:

$$\dot{\mathcal{X}} = F(\mathcal{X} - \overline{\mathcal{X}}) + G\dot{\mathbf{w}} \quad (\text{B.5})$$

where now

$$F = \begin{pmatrix} 0 & I \\ F_1 & F_2 \end{pmatrix} \quad \text{and} \quad G = \begin{pmatrix} 0 \\ G_0 \end{pmatrix}. \quad (\text{B.6})$$

This normal form is consistent with a state-space representation in terms of position and velocity:

$$\mathcal{X} = \begin{pmatrix} \mathbf{X} \\ \dot{\mathbf{X}} \end{pmatrix}.$$

The white noise $\dot{\mathbf{w}}$ can be interpreted mechanically as a (generalised) force applied to a particle whose configuration is $\mathbf{X}$. Now, from (B.3),

$$\mathcal{X}(t_k) - \overline{\mathcal{X}} = A'(\mathcal{X}(t_{k-1}) - \overline{\mathcal{X}}) + B'\mathbf{w}_k$$

$$\text{where } A' = \exp F\tau,$$

which can be written in terms of its submatrices as

$$A' = \begin{pmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{pmatrix}.$$

The matrix A' does not yet conform to the normal form (9.18) on page 204 for discrete coefficients A , which would require $A'_{11} = 0$ and $A'_{12} = I$. To reach the normal form, a coordinate transformation $\mathcal{X} \rightarrow M\mathcal{X}$ with

$$M = \begin{pmatrix} I & 0 \\ A'_{11} & A'_{12} \end{pmatrix}$$

must be applied. The transformed process has $A = M^{-1}A'M$ which is in normal form, and $B = M^{-1}B'$ which is only approximately (for small τ) in the normal form for B in (9.18) on page 204.

The inverse transformation, obtaining continuous parameters F and G from discrete ones A and B requires a matrix logarithm

$$F' = \frac{1}{\tau} \log A$$

followed by a coordinate change, similar to the one above, to reach the normal form.

Power spectrum

The expression (9.13) on page 200 for a second-order power spectrum is obtained from (B.5) using a frequency analysis similar to the one in the first-order case.

B.3 Accuracy of learning

The claims concerning accuracy of learning, stated in chapter 11 on page 241, are justified here. First the proportional error in the discrete dynamical parameters a_1 , a_2 and b_0 is obtained. Then this is used to derive the proportional error of the underlying continuous parameters f , β and $\bar{\rho}$.

Discrete analysis

The error of estimators $\hat{a}_1$, $\hat{a}_2$ and $\hat{b}_0$ is derived from the Fisher information measure for a maximum likelihood estimator (Kendall and Stuart, 1979). Asymptotically, for large M ,

$$\mathcal{V}[b_0|\hat{b}_0]^{-1} = -\mathcal{E} \left[\partial^2 L / \partial b_0^2 \right],$$

where L is the log-likelihood function (11.1) on page 237. Using (11.2) on page 237, this gives

$$\mathcal{V}[b_0|\hat{b}_0] = \frac{\hat{b}_0^2}{2(M-2)}$$

so that the proportional error in $\hat{b}_0$, denoted $\Delta\hat{b}_0$, is

$$\Delta\hat{b}_0 = \frac{\sqrt{\mathcal{V}[b_0|\hat{b}_0]}}{\hat{b}_0} = \frac{1}{\sqrt{2(M-2)}}. \quad (\text{B.7})$$

Applying a similar analysis to the vector $\mathbf{a} = (a_1, a_2)^T$ obtains error variances and covariances for a_1 and a_2 . This gives

$$\begin{aligned} \mathcal{V}[\mathbf{a}|\hat{\mathbf{a}}]^{-1} &= -\mathcal{E} [\partial^2 L / \partial \mathbf{a}^2] \\ &= \mathcal{E} \left[\frac{1}{\hat{b}_0^2} \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \right] \\ &= (M-2) \mathcal{E} \left[\frac{1}{\hat{b}_0^2} \mathcal{X}(t) \mathcal{X}(t)^T \right] \\ &= (M-2) \frac{1}{\hat{b}_0^2} \mathcal{P}_\infty \end{aligned}$$

where r_{ij} are the auto-correlation coefficients from the learning algorithm of figure 11.2 on page 238. Covariance $\mathcal{P}_\infty$ is obtained as the steady-state solution for $\mathcal{P}$ in the state equation (9.19) on page 204 for the process. After some manipulation, this gives

$$\mathcal{V}[\mathbf{a}|\hat{\mathbf{a}}] = \frac{1 + \hat{a}_2}{M-2} \begin{pmatrix} 1 - \hat{a}_2 & -\hat{a}_1 \\ -\hat{a}_1 & 1 - \hat{a}_2 \end{pmatrix}. \quad (\text{B.8})$$

For all 3 parameters a_1 , a_2 and b_0 it seems that estimator error $\propto 1/\sqrt{M-2}$ so that a typical training sequence of 1000 video fields should lead to error of just a few percent. This is a little misleading however because small changes in a_1 and a_2 can have a substantial effect on the ARP model. Looking at continuous parameters β , f and $\bar{\rho}$ gives a clearer picture.

Continuous analysis

Making the assumption that $\beta\tau \ll 1$ then, from (9.25) on page 206,

$$\beta\tau \approx \frac{1}{2}(1 + a_2)$$

so that, using (B.8),

$$\mathcal{V}[\beta\tau] \approx \frac{1}{4}\mathcal{V}[a_2] \approx \frac{1}{M-2}\hat{\beta}\tau$$

and finally

$$\Delta\hat{\beta} = \frac{\sqrt{\mathcal{V}[\beta]}}{\hat{\beta}} = \frac{1}{\sqrt{\hat{\beta}\tau}} \frac{1}{\sqrt{M-2}} \approx \frac{1}{\sqrt{\hat{\beta}T}},$$

as claimed in chapter 11. A similar analysis for $\hat{f}$ shows that

$$\Delta\hat{f} \equiv \frac{\sqrt{\mathcal{V}[f]}}{\hat{\beta}} = \frac{1}{2\pi} \frac{1}{\sqrt{\hat{\beta}T}}$$

Note that the error $\Delta\hat{f}$ in the estimated frequency is defined here relative to $\hat{\beta}$. Finally, it remains to establish an error bound for the estimated value of $\bar{\rho}$. From (9.27) on page 206, and given that $\Delta\hat{b}_0$ (see above) can be neglected,

$$\Delta\hat{\rho} = \frac{1}{2}\Delta\hat{\beta} \left(1 + \frac{\hat{\beta}}{\pi\hat{f}\sin 2\pi\hat{f}\tau} \right).$$

Appendix C

Further shape-space models

C.1 Recursive synthesis of shape-spaces

Chapter 4 ended with a discussion about shape-spaces for articulated motion and a summary comparing, for various hinged objects, the number of degrees of freedom of the object and the dimension of linear shape-space needed to represent its motion. There is a powerful general rule, presented here, for building up the dimension of a shape-space as articulated components are tacked onto a body. The dimension is built up recursively, tacking one component on at a time. It will be convenient to write the equation of the curve in *homogeneous coordinates*, a standard geometric tool in graphics and computer vision (Faugeras, 1993; Foley et al., 1990), giving

$$\mathbf{r}_h(s) = \begin{pmatrix} x(s) \\ y(s) \\ 1 \end{pmatrix}$$

so that the curve ranges over the shape-space swept out by

$$\mathbf{r}_h(s) = T\bar{\mathbf{r}}_h(s)$$

where the template $\bar{\mathbf{r}}_h(s)$ (also in homogeneous coordinates) may be either two-dimensional or three-dimensional as appropriate and T is a linear transformation. For example, the Euclidean similarities discussed earlier are represented by the following

transformation T :

$$T\mathbf{r} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} X_3 & -X_4 \\ X_4 & X_3 \end{pmatrix} \mathbf{r}$$

which can conveniently be written as a 3×3 matrix in homogeneous coordinates:

$$T = \begin{pmatrix} X_3 & -X_4 & X_1 \\ X_4 & X_3 & X_2 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{C.1})$$

This is appropriate, of course, for a body moving rigidly in the plane. More generally, it applies to the end link of a series of hinged links attached to a base that moves rigidly in the plane.

The new component is attached so that it is free to be acted on by transformations T' relative to the end body. For example, a simple planar hinge is represented in homogeneous coordinates by

$$T'(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Incremental rule: now the general rule can be stated, that when the hinged component is added, the dimension of the shape-space is increased by:

$$\dim\{T, TT'\}^+ - \dim\{T\}^+, \quad (\text{C.2})$$

where $\{T\}^+$ is the vector space of transformations T and $\{T_1, T_2\}^+$ denotes the vector space spanned by the two transformations taken jointly (simply concatenating the elements of T_1 and T_2 into one vector). To make this clear, we will work through the rule using the examples for T and T' given above of a planar base element moving rigidly in the plane with a single hinged component.

Planar rigid body with hinged appendage

First of all, $\dim\{T\}^+ = 4$, clearly, since T has 4 independent linear parameters $X_1, \dots, X_4$. Next we compute TT' :

$$TT' = \begin{pmatrix} \gamma & -\delta & X_1 \\ \delta & \gamma & X_2 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{C.3})$$

where $\gamma = X_3 \cos \theta - X_4 \sin \theta$ and $\delta = X_4 \cos \theta + X_3 \sin \theta$ so now

$$\dim\{T, TT'\}^+ = \dim\{X_1, X_2, X_3, X_4, \gamma, \delta\}^+ = 6$$

so that adding the hinge increases the dimension of shape-space by

$$\dim\{T, TT'\}^+ - \dim\{T\}^+ = 6 - 4 = 2$$

and the dimension of the new space is increased from 4 to 6.

Further hinged appendages

Now suppose we want to add a further hinged appendage. If it is added to the main body (figure C.1a), the argument above is unchanged (it is not in the least affected by the existence of the previous appendage) and the increase in dimension is still 2. Now the total dimension of the shape-space increases to 8. If instead the new appendage is tacked onto the end of the previous appendage (figure C.1b) we can apply the general method as follows. The matrix TT' in (C.3) above becomes the new T , and now

$$T' \equiv T'(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where ϕ is the angle of the latest appendage. Now the argument proceeds exactly as for the addition of the first appendage except that we have γ, δ and ϕ where before we had X_3, X_4 and θ , so again the subspace dimension is increased by 2. Alternatively, a quicker way to get to the same conclusion, is simply to note that the shape-space of the

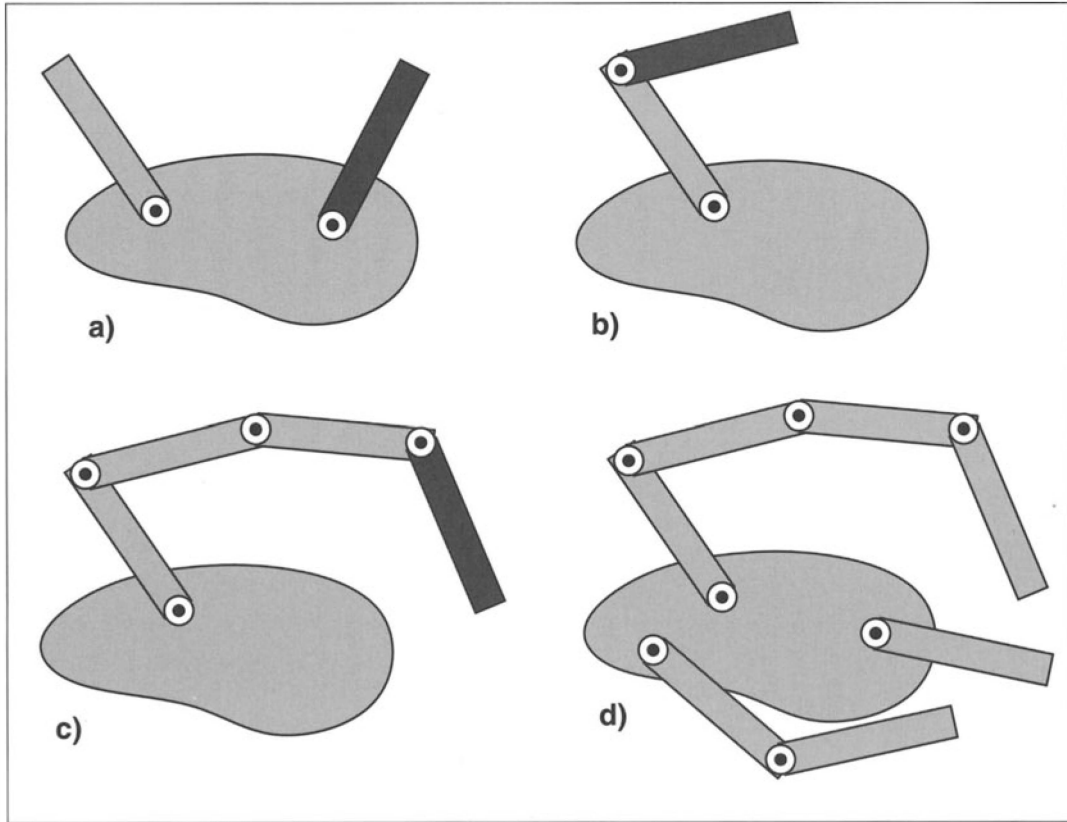


Figure C.1: Hinged appendages *Hinging an appendage onto: a) the main body; b) the end of another appendage; c) the end of a chain of appendages. d) A general planar articulated body consisting of a base in rigid motion with n hinged bodies.*

first appendage, considered in isolation, is the space of Euclidean similarities. (It can execute any rigid motion given that it is hinged to a base that can execute any rigid motion.) Therefore we are simply solving again the problem of computing the increase in shape-space dimension when an appendage is hinged to a base whose shape-space is the Euclidean similarities. Clearly we could continue to add appendages to a chain (figure C.1c), adding 2 to the shape-space dimension each time.

Now a simple inductive argument shows the following rule. A rigid planar body with n hinged appendages (figure C.1d) has a shape-space with dimension $4 + 2n$.

This is true regardless of how the hinges are arranged, provided there is no *closed* kinematic chain (sequence of hinged bodies forming a loop).

Adding telescopic appendages

How is the dimension of shape-space affected if an appendage is added with a “prismatic” or telescopic joint? In that case

$$T' \equiv T'(d) = \begin{pmatrix} 1 & 0 & d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where d is the variable length by which the joint is extended and taking T for Euclidean similarities of the base object as in (C.1), gives

$$TT' = \begin{pmatrix} X_3 & -X_4 & X_3d + X_1 \\ X_4 & X_3 & X_4d + X_2 \\ 0 & 0 & 1 \end{pmatrix},$$

so that

$$\dim(\{T, TT'\}^+) = \dim(\{X_1, X_2, X_3, X_4, X_3d, X_4d\}^+) = 6$$

(since X_3d, X_4d extend the basis by two elements — even though there is only one new degree of freedom d , it appears non-linearly and requires two degrees of freedom to represent linearly). Just as in the hinged case therefore, each telescopic appendage to a rigid body also raises the dimension of shape-space by two.

Planar body with co-planar appendage, in three dimensions

From earlier discussion in chapter 4, we know that images of a planar body in three dimensions form an affine space, so that $\{T\}^+$ is the usual 6-dimensional planar affine space. Unfortunately, unlike rigid planar bodies in which hinged appendages cost only 2 degrees of freedom each, in the planar affine case they come relatively expensively. Each requires 4 degrees of freedom so that insisting on linear parameterisation is relatively costly.

The argument, applying the incremental rule (C.2) is as follows. Transformations are

$$T = \begin{pmatrix} X_3 & X_6 & X_1 \\ X_5 & X_4 & X_2 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad T' = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and after a little calculation

$$\dim(\{T, TT'\}^+) = \dim(\{X_1, X_2, ((X_n \cos \phi, X_n \sin \phi), n = 3, \dots, 6)\}^+) = 10,$$

an increase of 4 over the 6 affine degrees of freedom.

Proof of incremental rule

For completeness, a proof of the rule (C.2) for incrementing the dimension of shape-space is included here. Consider a base shape γ which we would normally represent by a template either as a parameterised curve $\bar{\mathbf{r}}(s)$ in two dimensions or $\bar{\mathbf{R}}(s)$ in three dimensions, or as a control point vector $\bar{\mathbf{Q}}$ in two or three dimensions. It is subject to linear transformations T onto the image plane, parameterised (not necessarily linearly) by a parameter set λ , giving a set of image shapes $\{T(\lambda)\gamma, \lambda \in \Lambda\}$. This set spans a vector space denoted by the closure $\{T(\lambda)\gamma, \lambda \in \Lambda\}^+$ which generally (for non-degenerate γ) is isomorphic to the closure of the space of transformations $\{T(\lambda), \lambda \in \Lambda\}^+$, regardless of the particular shape γ .

Next, it is assumed that the body to which an appendage is about to be added is already articulated so that, already attached to the base, are a set of components transformed relative to the base by a set of linear transformations $T_n(\lambda_n)$, $\lambda_n \in \Lambda_n$ for $n = 1 \dots N$. The n^{th} component is thus transformed into the image plane by the transformation $TT_n(\lambda_n)$. Finally, the appended component is attached via $T'(\lambda')$, $\lambda' \in \Lambda'$. It is assumed that the parameters $\lambda_1, \dots, \lambda_N, \lambda'$ are all independent — the hinging/telescopic actions of the individual components are not coupled and this is where closed kinematic chains are excluded. We also assume that all components are non-degenerate so that we can continue to consider the transformations only and drop any reference to the component shapes themselves.

The independence of parameters for components means that

$$\dim\{T, TT'\}^+ + \dim\{T, TT_1, \dots, TT_N\}^+ = \dim\{T, TT_1, \dots, TT_N, TT'\}^+ + \dim\{T\}^+$$

(omitting for simplicity explicit reference to parameters $\lambda, \lambda_1 \dots \lambda_n$ and λ') and this is simply rearranged into a formula for the increase in shape-space dimension:

$$\dim\{T, TT_1, \dots, TT_N, TT'\}^+ - \dim\{T, TT_1, \dots, TT_N\}^+ = \dim\{T, TT'\}^+ - \dim\{T\}^+,$$

the right hand side of which is the required formula (C.2), simplified in that it involves only the base and the new component.

Silhouettes

For smooth silhouette curves, it can be shown that a shape-space of dimension 11 is appropriate. This shape-space representation of the curve is an approximation, valid for sufficiently small changes of viewpoint. The proof of this result follows from results in the computer vision literature about the projection of silhouettes into images (Giblin and Weiss, 1987; Blake and Cipolla, 1990; Vaillant, 1990; Koenderink, 1990; Cipolla and Blake, 1992b).

Glossary of notation

SYMBOL	MEANING	see page
$\hat{x}$	an <i>estimate</i> for the quantity x	
$\tilde{x}$	a <i>prediction</i> of the quantity x	
$\dot{x}$	<i>temporal derivative</i> of x	
$\mathbf{x} \cdot \mathbf{y}$	scalar product of vectors $\mathbf{x}, \mathbf{y}$	
$\ \cdot\ $	norm for functions/curves	47,58
$\langle \cdot, \cdot \rangle$	inner product for functions/curves	47,58
$A \otimes B$	Kronecker product of matrices A, B	282
T	superscript denoting vector/matrix transpose	
$\mathbf{0}$	vector of zeros, of length N_B	
$\mathbf{1}$	vector of ones, of length N_B	
A	deterministic coefficient matrix in discrete dynamical model	204
A_1, A_2	components of A	204
$\mathbf{B}(s)$	vector of B-spline blending functions	44
B_0	stochastic coefficients in second-order discrete dynamical model	204
B	stochastic coefficient matrix in discrete dynamical model	193, 204
$\mathcal{B}$	metric matrix for B-spline functions	50
$B_m(s)$	m th B-spline blending function	43
C	covariance coefficient $C = BB^T$	242
d	order of spline polynomial	
$\mathcal{E}[Y]$	expectation of a random variable Y	294
H	observation matrix for Kalman filter	216

SYMBOL	MEANING	see page
$\mathcal{H}$	metric matrix for curves in shape-space $\mathcal{S}$	79
$\mathbf{h}(s)$	image measurement matrix for recursive curve-fitting	124
I_r	$r \times r$ identity matrix	
$K(t), \mathcal{K}(t)$	Kalman gain	216
k	index for discrete time $t_k = k\tau$	
κ	length factor for search lines	174
L	no. of spans on B-spline curve	45
M	no. of frames in a training sequence $\mathbf{X}_1, \dots, \mathbf{X}_M$	175
N	no. of sampled image features along a B-spline curve	124
N_B	no. of control points on B-spline curve	43
N_Q	dimension of spline space	65
N_X	dimension of configuration space (shape-space) $\mathcal{S}$	69
$\mathcal{N}(\overline{\mathbf{X}}, P)$	multi-variate Gaussian (normal) distribution	295
$\mathbf{n}(s, t)$	image-curve normal	122
ν_i	innovation due to i th image feature	124
P, P', P''	covariance components for shape $\hat{\mathbf{X}}(t)$	218
$\mathcal{P}(t)$	covariance of state $\hat{\mathcal{X}}(t)$	204, 213
$\tilde{\mathcal{P}}(t_k)$	covariance of predicted state-vector $\tilde{\mathcal{X}}(t_k)$	216
$\mathbf{q}_n$	control points for spline curve	53
$\mathbf{Q}$	vector of control points	58
$\mathbf{Q}^x$	vector of x -coordinates of control points	44, 58
$\mathbf{Q}^y$	vector of y -coordinates of control points	58

SYMBOL	MEANING	see page
$\mathbf{Q}_0$	control points of template curve	74
R	rotation matrix	283
$\mathbf{R}(s)$	space-curve (3D)	83
R_{ij}, R'_{ij}	auto-correlation coefficients of training sequence	244
R_i	training-sequence sums	244
$\mathbf{r}(s)$	image curve (2D)	53
$\rho_0(s)$	root-mean-square displacement at s on a curve	161
$\bar{\rho}_0$	root-mean-square displacement of curve	161
S, S_i	statistical information matrix	127
$\mathcal{S}_Q$	space of spline curves $\mathbf{Q}$	58
$\mathcal{S}$	shape-space	69
s	spatial parameter for curve	
σ	σ^2 is variance of image measurement process	127,169
t	time parameter	
T	duration $T = M\tau$ of an image sequence	
τ	sample interval for image capture	
$U(s)$	matrix mapping control point vector $\mathbf{Q}$ to image curve $\mathbf{r}(s)$	58
$\mathbf{u}$	translation vector	76
$\mathcal{U}$	metric matrix for B-spline parametric curves	59
$\mathcal{V}[Y]$	variance of a random variable Y	294
W	shape-matrix mapping from configuration $\mathbf{X}$ to control point vector $\mathbf{Q}$	74
$W^\dagger$	pseudo-inverse of mapping W	74
$\mathbf{w}_k$	discrete noise: vector of independent standard normal variables	193

SYMBOL	MEANING	see page
$\mathbf{X}$	curve shape-vector	74
$\bar{\mathbf{X}}$	mean value of curve configuration	160
$\mathcal{X}$	state-vector for image-based dynamics at discrete time k	204
$\hat{\mathcal{X}}(t_k)$	estimated state-vector from a tracker	214
$\tilde{\mathcal{X}}(t_k)$	predicted state-vector in a tracker	216
$\bar{\mathcal{X}}$	mean value of state $\mathcal{X}$	204
$\underline{\mathcal{X}}(t_k)$	history of states $\mathcal{X}(t_1), \dots, \mathcal{X}(t_k)$	260
$\mathbf{Z}(t_k)$	aggregated vector of visual measurements at time $t = k\tau$	127
$\underline{\mathbf{Z}}(t_k)$	measurement history $\{\mathbf{Z}(t_1), \dots, \mathbf{Z}(t_k)\}$ up to time $t = k\tau$	213

Bibliography

Adjoudani, A. and Benoit, C. (1995). On the integration of auditory and visual parameters in an HMM-based ASR. In *Proceedings NATO ASI Conference on Speechreading by Man and Machine: Models, Systems and Applications*, 461–472. NATO Scientific Affairs Division.

Allen, P., Yoshimi, B., and Timcenko, A. (1991). Real-time visual servoing. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1, 851–856.

Aloimonos, J. (1990). Perspective approximations. *J. Image and Vision Computing*, 8, 177–192.

Aloimonos, J. (1993). *Active Perception*. Erlbaum.

Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1987). Active vision. In *Proc. 1st Int. Conf. on Computer Vision*, 35–54.

Amini, A., Owen, R., Staib, L., Anandan, P., and Duncan, J. (1991). *Non-rigid motion models for tracking the left ventricular wall*. Lecture notes in computer science: Information processing in medical images. Springer-Verlag.

Amini, A., Tehrani, S., and Weymouth, T. (1988). Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Proc. 2nd Int. Conf. on Computer Vision*, 95–99.

Anderson, B. and Moore, J. (1979). *Optimal filtering*. Prentice Hall.

Arbogast, E. and Mohr, R. (1990). 3D structure inference from images sequences. In Baird, H., editor, *Proceedings of the Syntactical and Structural Pattern Recognition Workshop*, 21–37, Murray-Hill, NJ.

- Astrom, K. (1970). *Introduction to stochastic control theory*. Academic Press.
- Astrom, K. and Wittenmark, B. (1984). *Computer Controlled Systems*. Addison Wesley.
- Ayache, N., Cohen, I., and Herlin, I. (1992). Medical image tracking. In Blake, A. and Yuille, A., editors, *Active Vision*, 285–302. MIT.
- Ayache, N. and Faugeras, O. (1987). Building, registration and fusing noisy visual maps. In *Proc. 1st Int. Conf. on Computer Vision*, London.
- Ayache, N. and Faverjon, B. (1987). Efficient registration of stereo images by matching graph descriptions of edge segments. *Int. J. Computer Vision*, 107–131.
- Azarbayejani, A., Starner, T., Horowitz, B., and Pentland, A. (1993). Visually controlled graphics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15, 6, 602–605.
- Bajcsy, R. (1988). Active perception. In *Proc. IEEE*, 76, 996–1005.
- Baker, H. and Binford, T. (1981). Depth from edge and intensity based stereo. In *Proc. Int. Joint Conf. Artificial Intelligence*, 631–636.
- Ballard, D. (1981). Generalising the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 12, 2, 111–122.
- Ballard, D. and Brown, C. (1982). *Computer Vision*. Prentice–Hall, New Jersey.
- Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press.
- Barnett, S. (1990). *Matrices: Methods and Applications*. Oxford University Press.
- Bartels, R., Beatty, J., and Barsky, B. (1987). *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann.
- Bascle, B. and Deriche, R. (1995). Region tracking through image sequences. In *Proc. 5th Int. Conf. on Computer Vision*, 302–307, Boston.
- Baumberg, A. and Hogg, D. (1994). Learning flexible models from image sequences. In Eklundh, J.-O., editor, *Proc. 3rd European Conf. Computer Vision*, 299–308. Springer–Verlag.

- Baumberg, A. and Hogg, D. (1995a). An adaptive eigenshape model. *Proc. British Machine Vision Conf.*, 87–96.
- Baumberg, A. and Hogg, D. (1995b). Generating spatiotemporal models from examples. In *Proc. British Machine Vision Conf.*, 2, 413–422.
- Belhumeur, P. (1993). A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *Proc. 4th Int. Conf. on Computer Vision*, 431–438.
- Belhumeur, P., Hespanha, J., and Kriegman, D. (1996). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. In *Proc. 4th European Conf. Computer Vision*, number 800 in Lecture notes in computer science, 45–58. Springer-Verlag.
- Bellman, R. and Dreyfus, S. (1962). *Applied dynamic programming*. Princeton University Press, Princeton, U.S.
- Bennett, A. and Craw, I. (1991). Finding image features for deformable templates and detailed prior statistical knowledge. In *Proc. British Machine Vision Conf.*, 233–239.
- Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992a). Hierarchical model-based motion estimation. In *Proc. 2nd European Conf. Computer Vision*, 237–252.
- Bergen, J., Burt, P., Hingorani, R., and Peleg, S. (1992b). A three-frame algorithm for estimating two-component image motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14, 9, 886–896.
- Beymer, D. and Poggio, T. (1995). Face recognition from one example view. In *Proc. 5th Int. Conf. on Computer Vision*, 500–507.
- Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford.
- Black, M. and Jepson, A. (1996). Eigenttracking: robust matching and tracking of articulated objects using a view-based representation. In *Proc. 4th European Conf. Computer Vision*, 329–342.
- Black, M. and Yacoob, Y. (1995). Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proc. 5th Int. Conf. on Computer Vision*, 374–381.

- Blake, A. (1992). Computational modelling of hand-eye coordination. *Phil. Trans. R. Soc.* 337, 351–360.
- Blake, A. and Cipolla, R. (1990). Robust estimation of surface curvature from deformation of apparent contours. In Faugeras, O., editor, *Proc. 1st European Conf. Computer Vision*, 465–474. Springer-Verlag.
- Blake, A., Curwen, R., and Zisserman, A. (1993). A framework for spatio-temporal control in the tracking of visual contours. *Int. J. Computer Vision*, 11, 2, 127–145.
- Blake, A. and Isard, M. (1994). 3D position, attitude and shape input using video tracking of hands and lips. In *Proc. Siggraph*, 185–192. ACM.
- Blake, A., Isard, M., and Reynard, D. (1995). Learning to track the visual motion of contours. *J. Artificial Intelligence*, 78, 101–134.
- Blake, A. and Marinos, C. (1990). Shape from texture: estimation, isotropy and moments. *J. Artificial Intelligence*, 45, 323–380.
- Blake, A. and Yuille, A., editors (1992). *Active Vision*. MIT.
- Bobick, A. and Wilson, A. (1995). A state-based technique for the summarisation and recognition of gesture. In *Proc. 5th Int. Conf. on Computer Vision*, 382–388.
- Bolles, R., Baker, H., and Marimont, D. (1987). Epipolar-plane image analysis: An approach to determining structure. *Int. J. Computer Vision*, 1, 7–55.
- Bookstein, F. (1988). Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.
- Bracewell, R. (1978). *The Fourier transform and its applications*. McGraw-Hill.
- Bray, A. (1990). Tracking objects using image disparities. *J. Image and Vision Computing*, 8, 1, 4–9.
- Bregler, C. and Konig, Y. (1994). Eigenlips for robust speech recognition. In *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 669–672, Adelaide.
- Bregler, C. and Omohundro, S. (1994). Surface learning with applications to lipreading. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6*, 6. Morgan Kaufmann Publishers.

- Bregler, C. and Omohundro, S. (1995). Nonlinear manifold learning for visual speech recognition. In *Proc. 5th Int. Conf. on Computer Vision*, 494–499, Boston.
- Broida, T. and Chellappa, R. (1986). Estimation of object motion parameters from noisy images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8, 90–99.
- Brown, C., Coombs, D., and Soong, J. (1992). Real-time smooth pursuit tracking. In Blake, A. and Yuille, A., editors, *Active Vision*, 123–136. MIT.
- Brown, C. and Terzopoulos, D., editors (1994). *Real-time computer vision*. Cambridge.
- Bruce, J. and Giblin, P. (1984). *Curves and Singularities*. Cambridge.
- Bucy, R. (1969). Bayes theorem and digital realizations for non-linear filters. *J. Astronautical Sciences*, 17, 2, 80–94.
- Bulthoff, H., Little, J., and Poggio, T. (1989). A parallel algorithm for real-time computation of optical flow. *Nature*, 337, 9, 549–553.
- Burr, D. (1981). Elastic matching of line drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3, 6, 708–713.
- Burt, P. (1983). Fast algorithms for estimating local image properties. *Computer Vision, Graphics and Image Processing*, 21, 368–382.
- Buxton, B. and Buxton, H. (1983). Monocular depth perception from optical flow by space time signal processing. *Proc. Royal Society of London*, B 218, 27–47.
- Buxton, B. and Buxton, H. (1984). Computation of optic flow from the motion of edge features in image sequences. *J. Image and Vision Computing*, 2, 59–75.
- Campbell, L. and Bobick, A. (1995). Recognition of human body motion using phase space constraints. In *Proc. 5th Int. Conf. on Computer Vision*, 624–630.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8, 6, 679–698.
- Cipolla, R. and Blake, A. (1990). The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, 616–625.

Cipolla, R. and Blake, A. (1992a). Motion planning using image divergence and deformation. In Blake, A. and Yuille, A., editors, *Active Vision*, 39–58. MIT.

Cipolla, R. and Blake, A. (1992b). Surface shape and the deformation of apparent contours. *Int. J. Computer Vision*, 9, 2, 83–112.

Cohen, L. (1991). On active contour models and balloons. *CVGIP: Image Understanding*, 53, 2, 211–218.

Cootes, T., Hill, A., Taylor, C., and Haslam, J. (1994). The use of active shape models for locating structures in medical images. *J. Image and Vision Computing*, 12, 6, 355–366.

Cootes, T. and Taylor, C. (1992). Active shape models. In *Proc. British Machine Vision Conf.*, 265–275.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61, 1, 38–59.

Cootes, T., Taylor, C., Lanitis, A., Cooper, D., and Graham, J. (1993). Building and using flexible models incorporating grey-level information. In *Proc. 4th Int. Conf. on Computer Vision*, 242–246.

Cox, I. (1993). A review of statistical data association techniques for motion correspondence. *Int. J. Computer Vision*, 10, 1, 53–66.

Craig, J. (1986). *Introduction to robotics: mechanics and control*. Addison-Wesley.

Crisman, J. D. (1992). Color region tracking for vehicle guidance. In Blake, A. and Yuille, A., editors, *Active Vision*, chapter 7. The MIT Press.

Curwen, R. (1993). *Dynamic and Adaptive Contours*. PhD thesis, University of Oxford.

Curwen, R. and Blake, A. (1992). Dynamic contours: real-time active splines. In Blake, A. and Yuille, A., editors, *Active Vision*, 39–58. MIT.

Curwen, R., Blake, A., and Cipolla, R. (1991). Parallel implementation of Lagrangian dynamics for real-time snakes. In *Proc. British Machine Vision Conf.*, 29–36.

- Davidson, C. and Blake, A. (1998). Error-tolerant visual planning of planar grasp. In *Proc. 6th Int. Conf. on Computer Vision*.
- de Boor, C. (1978). *A practical guide to splines*. Springer-Verlag, New York.
- Demey, S., Zisserman, A., and Beardsley, P. (1992). Affine and projective structure from motion. In *Proc. British Machine Vision Conf.*
- Dempster, A., Laird, M., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc., B*, 39, 1–38.
- Deriche, R. and Faugeras, O. (1990). Tracking line segments. In Faugeras, O., editor, *Proc. 1st European Conf. Computer Vision*, 259–268. Springer-Verlag.
- Dickmanns, E. (1992). Expectation-based dynamic scene understanding. In Blake, A. and Yuille, A., editors, *Active Vision*, 303–336. MIT.
- Dickmanns, E. and Graefe, V. (1988a). Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1, 241–261.
- Dickmanns, E. and Graefe, V. (1988b). Dynamic monocular machine vision. *Machine Vision and Applications*, 1, 223–240.
- Dodd, B. and Campbell, R. (1987). *Hearing By Eye: The Psychology of Lip Reading*. Erlbaum.
- Dreschler, L. and Nagel, H. (1981). Volumetric model and 3D trajectory of a moving car derived from monocular TV-frame sequence of a street scene. In *Proc. Int. Joint Conf. Artificial Intelligence*, 692–697.
- Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons.
- Durrant-Whyte, H. (1988). Uncertain geometry in robotics. *IEEE J. of Robotics and Automation*, 4, 1, 23–31.
- Enkelmann, W. (1986). Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. In *Proc. Conf. Computer Vision and Pattern Recognition*, 81–87.

- Essa, I. and Pentland, A. (1995). Facial expression recognition using a dynamic model and motion energy. In *Proc. 5th Int. Conf. on Computer Vision*, 360–367.
- Faugeras, O. (1993). *3D Computer Vision*. MIT Press.
- Faugeras, O. and Hebert, M. (1986). The representation, recognition, and locating of 3D objects. *Int. J. Robotics Research*, 5, 3, 27–52.
- Faux, I. and Pratt, M. (1979). *Computational Geometry for Design and Manufacture*. Ellis-Horwood.
- Faverjon, B. and Ponce, J. (1991). On computing two-finger force-closure grasps of curved 2D objects. In *Proc. IEEE Int. Conf. Robotics and Automation*, 3, 424–429.
- Ferrier, N., Rowe, S., and Blake, A. (1994). Real-time traffic monitoring. In *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, 81–88. IEEE.
- Finn, E. and Montgomery, A. (1988). Automatic optically based recognition of speech. *Pattern Recognition Letters*, 8, 3, 159–164.
- Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24, 381–95.
- Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22, 1.
- Foley, J., van Dam, A., Feiner, S., and Hughes, J. (1990). *Computer Graphics: Principles and Practice*, chapter 13, 563–604. Addison-Wesley.
- Forsyth, D., Mundy, J., Zisserman, A., and Brown, C. (1990). Projectively invariant representations using implicit algebraic curves. In Faugeras, O., editor, *Proc. 1st European Conf. Computer Vision*, 427–436. Springer-Verlag.
- Freeman, W. and Tenenbaum, J. (1997). Learning bilinear models for two-factor problems in vision. In *Proc. Conf. Computer Vision and Pattern Recognition*, 554–560.
- Gee, A. and Cipolla, R. (1994). Determining the gaze of faces in images. *J. Image and Vision Computing*, 12, 10, 639–647.

- Gee, A. and Cipolla, R. (1996). Fast visual tracking by temporal consensus. *J. Image and Vision Computing*, 14, 2, 105–114.
- Gelb, A., editor (1974). *Applied Optimal Estimation*. MIT Press, Cambridge, MA.
- Gelfand, A. and Smith, A. (1990). Sampling-based approaches to computing marginal densities. *J. Am. Statistical Assoc.*, 85, 410, 398–409.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6, 6, 721–741.
- Gennery, D. (1992). Visual tracking of known three-dimensional objects. *Int. J. Computer Vision*, 7, 3, 243–270.
- Giblin, P. and Weiss, R. (1987). Reconstruction of surfaces from profiles. In *Proc. 1st Int. Conf. on Computer Vision*, 136–144, London.
- Golub, G. and van Loan, C. (1989). *Matrix computations*. Johns Hopkins.
- Goncalves, L., di Bernardo, E., Ursella, E., and Perona, P. (1995). Monocular tracking of the human arm in 3d. In *Proc. 5th Int. Conf. on Computer Vision*, 764–770.
- Gonzales, R. and Wintz, P. (1987). *Digital Image Processing*. Addison-Wesley.
- Goodwin, C. and Sin, K. (1984). *Adaptive filtering prediction and control*. Prentice-Hall.
- Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F*, 140, 2, 107–113.
- Grenander, U. (1976–1981). *Lectures in Pattern Theory I, II and III*. Springer.
- Grenander, U., Chow, Y., and Keenan, D. (1991). *HANDS. A Pattern Theoretical Study of Biological Shapes*. Springer-Verlag. New York.
- Grimson, W. (1981). *From Images to Surfaces*. MIT Press, Cambridge, USA.
- Grimson, W. (1985). Computational experiments with a feature based stereo algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7, 1, 17–34.

Grimson, W. (1990). *Object recognition by computer*. MIT Press.

Grimson, W. and Lozano-Perez, T. (1984). Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Research*, 5, 3, 3–34.

Grimson, W., Lozano-Perez, T., Wells, W., Ettinger, G., White, S., and Kikinis, R. (1994). An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. In *Proc. Conf. Computer Vision and Pattern Recognition*, 430–436, Seattle, WA.

Hager, G. (1990). *Sensor fusion and planning: a computational approach*. Kluwer Academic Publishers.

Hager, G. and Belhumeur, P. (1996). Real-time tracking of image regions with changes in geometry and illumination. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 403–410.

Hallam, J. (1983). Resolving observer motion by object tracking. In *Proc. Int. Joint Conf. Artificial Intelligence*, 2, 792–798.

Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. (1995). *Robust Statistics*. John Wiley and Sons, New York.

Haralick, R. (1980). Edge and region analysis for digital image data. *Computer Graphics and Image Processing*, 12, 1, 60–73.

Harris, C. (1990). Structure from motion under orthographic projection. In Faugeras, O., editor, *Proc. 1st European Conf. Computer Vision*, 118–123. Springer-Verlag.

Harris, C. (1992a). Geometry from visual motion. In Blake, A. and Yuille, A., editors, *Active Vision*, 263–284. MIT.

Harris, C. (1992b). Tracking with rigid models. In Blake, A. and Yuille, A., editors, *Active Vision*, 59–74. MIT.

Harris, C. and Stennett, C. (1990). Rapid – a video-rate object tracker. In *Proc. British Machine Vision Conf.*, 73–78.

Heeger, D. (1987). Optical flow from spatiotemporal filters. In *Proc. 1st Int. Conf. on Computer Vision*, 181–190.

- Heuring, J. and Murray, D. (1996). Visual head tracking and slaving for visual telepresence. In *Proc. IEEE Int Conf. Robotics and Automation, Minneapolis, May 1996*, Los Alamitos, CA. IEEE Computer Society Press.
- Hildreth, E. (1983). *The Measurement of Visual Motion*. MIT Press.
- Hinton, G., Williams, C., and Revow, M. (1992). Adaptive elastic models for hand-printed character recognition. *Advances in Neural Information Processing Systems*, 4.
- Hogg, D. (1983). Model-based vision: a program to see a walking person. *J. Image and Vision Computing*, 1, 1, 5–20.
- Horn, B. (1986). *Robot Vision*. McGraw-Hill, NY.
- Horn, B. and Schunk, B. (1981). Determining optical flow. *J. Artificial Intelligence*, 17, 185–203.
- Huang, X., Arika, Y., and Jack, M. (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh University Press.
- Huttenlocher, D., Noh, J., and Rucklidge, W. (1993). Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, 93–101.
- Ikeuchi, K. and Horn, B. (1981). Numerical shape from shading and occluding boundaries. *J. Artificial Intelligence*, 17, 141–184.
- Inoue, H. and Mizoguchi, H. (1985). A flexible multi window vision system for robots. In *Proc. 2nd Int. Symp. on Robotics Research*, 95–102.
- Isard, M. and Blake, A. (1998a). Condensation — conditional density propagation for visual tracking. *Int. J. Computer Vision*.
- Isard, M. and Blake, A. (1998b). A mixed-state Condensation tracker with automatic model switching. In *Proc. 6th Int. Conf. on Computer Vision*.
- Ivins, J. and Porrill, J. (1995). Active region models for segmenting textures and colours. *J. Image and Vision Computing*, 13, 5, 431–438.
- Jacobs, O. (1993). *Introduction to control theory*. Oxford University Press.

Jain, A. and Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24, 12, 1167–1186.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, 259–268.

Kaucic, R., Dalton, B., and Blake, A. (1996). Real-time liptracking for audio-visual speech recognition applications. In *Proc. 4th European Conf. Computer Vision*, 376–387, Cambridge, England.

Kendall, M. and Stuart, A. (1979). *The advanced theory of statistics, vol 2, inference and relationship*. Charles Griffing and Co Ltd, London.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5, 1, 1–25.

Kitchen, L. and Rosenfeld, A. (1982). Grey-level corner detection. *Pattern Recognition Letters*, 1, 95–102.

Koenderink, J. (1990). *Solid Shape*. MIT Press.

Koenderink, J. and van Doorn, A. (1991). Affine structure from motion. *J. Optical Soc. of America A.*, 8, 2, 337–385.

Koller, D., Weber, J., and Malik, J. (1994). Robust multiple car tracking with occlusion reasoning. In *Proc. 3rd European Conf. Computer Vision*, 189 – 196. Springer-Verlag.

Kreysig, E. (1988). *Advanced Engineering Mathematics*. Wiley.

Kutulakos, K. and Valliano, J. (1996). Non-euclidean object representations for calibration-free video overlay. In Ponce, J., Zisserman, A., and Hebert, M., editors, *Object Representation in Computer Vision II*, Lecture Notes in Computer Science, 381–401. Springer-Verlag.

Landau, L. and Lifshitz, E. (1972). *Mechanics and Electrodynamics*, 1 of *A Shorter Course of Theoretical Physics*. Pergamon Press.

Lanitis, A., Taylor, C., and Cootes, T. (1995). A unified approach to coding and interpreting face images. In *Proc. 5th Int. Conf. on Computer Vision*, 368–373.

- Lawn, J. and Cipolla, R. (1994). Robust egomotion estimation from affine motion parallax. In *Proc. 3rd European Conf. Computer Vision*, 205–210.
- Ljung, L. (1987). *System identification: theory for the user*. Prentice-Hall.
- Lowe, D. (1991). Fitting parameterised 3D models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13, 5, 441–450.
- Lowe, D. (1992). Robust model-based motion tracking through the integration of search and estimation. *Int. J. Computer Vision*, 8, 2, 113–122.
- Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conf. Artificial Intelligence*, 674–679.
- MacCormick, J. and Blake, A. (1998). A probabilistic contour discriminant for object localisation. In *Proc. 6th Int. Conf. on Computer Vision*.
- Marais, P., Guillemaud, R., Sakuma, M., Zisserman, A., and Brady, J. (1996). Visualising cerebral asymmetry. In *Visualisation in Biomedical Computing*, Lecture notes in Computer Science, 411–416. Springer-Verlag.
- Marchant, J. (1991). Intelligent machinery for agriculture. *Machine Vision and Applications*, 151, 177–186.
- Mardia, K., Ghali, N., Howes, M., Hainsworth, T., and Sheehy, N. (1993). Techniques for online gesture recognition. *J. Image and Vision Computing*, 11, 5, 283–294.
- Marr, D. (1982). *Vision*. Freeman, San Francisco.
- Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proc. Roy. Soc. London. B.*, 207, 187–217.
- Matthies, L., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. Computer Vision*, 3, 209–236.
- Mayhew, J., Zheng, Y., and Cornell, S. (1992). The adaptive control of a four-degrees-of-freedom stereo camera head. *Phil. Trans. R. Soc. Lond. B*, 337, 5, 315–326.
- Medioni, G. and Yasumoto, Y. (1986). Corner detection and curve representation using curve b-splines. In *Proc. Conf. Computer Vision and Pattern Recognition*, 764–769.

- Menet, S., Saint-Marc, P., and Medioni, G. (1990). B-snakes: Implementation and application to stereo. In *Proceedings DARPA*, 720–726.
- Montanari, U. (1971). On the optimal detection of curves in noisy pictures. *Commun. ACM*, 14, 5, 335–345.
- Mumford, D. (1996). Pattern theory: a unifying perspective. In Knill, D. and Richard, W., editors, *Perception as Bayesian inference*, 25–62. Cambridge University Press.
- Mundy, J. and Heller, A. (1990). The evolution and testing of a model-based object recognition system. In *Proc. 3rd Int. Conf. on Computer Vision*, 268–282.
- Mundy, J. and Zisserman, A. (1992). *Geometric invariance in computer vision*. MIT Press.
- Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3D objects from appearance. *Int. J. Computer Vision*, 14, 1, 5–24.
- Murray, D. and Basu, A. (1994). Motion tracking with an active camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16, 5, 449–459.
- Murray, D. and Buxton, B. (1990). *Experiments in the machine interpretation of visual motion*. MIT Press.
- Murray, D., Du, F., McLauchlan, P., Reid, I., Sharkey, P., and Brady, J. (1992). Design of stereo heads. In Blake, A. and Yuille, A., editors, *Active Vision*, 303–336. MIT.
- Murray, D., McLauchlan, P., Reid, I., and Sharkey, P. (1993). Reactions to peripheral image motion using a head/eye platform. In *Proc. 4th Int. Conf. on Computer Vision*, 403–411.
- Nagel, H. (1983). Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21, 85–117.
- Nagel, H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8, 5, 565–593.

- Noble, J. (1988). Finding corners. *J. Image and Vision Computing*, 6, 2, 121–128.
- North, B. and Blake, A. (1998). Using expectation-maximisation to learn a dynamical model for a tracker from measurement sequences. In *Proc. 6th Int. Conf. on Computer Vision*.
- O’Gorman, F. (1978). Edge detection using Walsh functions. *J. Artificial Intelligence*, 10, 215–223.
- Ohta, Y. and Kanade, T. (1985). Stereo by intra- and inter-scan line search using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7, 2, 139–154.
- Papanikolopoulos, N., Khosla, P., and Kanade, T. (1991). Vision and control techniques for robotic visual tracking. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1, 851–856.
- Papoulis, A. (1990). *Probability and Statistics*. Prentice-Hall.
- Papoulis, A. (1991). *Probability, random variables and stochastic processes*. McGraw-Hall.
- Pardey, J., Roberts, S., and Tarassenko, L. (1995). A review of parametric modelling techniques for EEG analysis. *Medical Engineering Physics*, 18, 1, 2–11.
- Pelizzari, C., Tan, K., Levin, D., Chen, G., and Balter, J. (1993). Interactive 3d patient — image registration. In *Proc. 13th Int. Conf. on Information Processing in Medical Imaging*, 132–141, Berlin, Germany.
- Pentland, A. and Horowitz, B. (1991). Recovery of nonrigid motion and structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7, 730–742.
- Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12, 7, 629–639.
- Petajan, E., Bischofy, B., Bodoff, D., and Brooke, N. (1988). An improved automatic lipreading system to enhance speech recognition. In Soloway, E., Frye, D., and Sheppard, S., editors, *Proc. Human Factors in Computing Systems*, 19–25. ACM.
- Petitjean, S., Ponce, J., and Kriegman, D. (1992). Computing exact aspect graphs of curved objects: Algebraic surfaces. *Int. J. Computer Vision*, 9, 231–255.

- Plá, F., Juste, F., Ferri, F., and Vicens, M. (1993). Colour segmentation based on a light reflection model to locate citrus fruits for robotic harvesting. *Computers and Electronics in Agriculture*, 9, 1, 53–70.
- Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularisation theory. *Nature*, 317, 314–319.
- Pollard, S., Mayhew, J., and Frisby, J. (1985). PMF: a stereo correspondence algorithm using a disparity gradient. *Perception*, 14, 449–470.
- Ponce, J., Burdick, J., and Rimon, E. (1995). Computing the immobilizing three-finger grasps of planar objects. In *Proc. of the 1995 Workshop on Computational Kinematics*, 281–300.
- Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1988). *Numerical Recipes in C*. Cambridge University Press.
- Rabiner, L. and Bing-Hwang, J. (1993). *Fundamentals of speech recognition*. Prentice-Hall.
- Ramer, E. (1975). The transformation of photographic images into stroke arrays. *IEEE Trans. CAS*, 22,3, 363–374.
- Rao, B., Durrant-Whyte, H., and Sheen, J. (1993). A fully decentralized multi-sensor system for tracking and surveillance. *Int. J. Robotics Research*, 12, 1, 20–44.
- Rao, C. (1973). *Linear Statistical Inference and Its Applications*. John Wiley and Sons, New York.
- Rehg, J. and Kanade, T. (1994). Visual tracking of high dof articulated structures: an application to human hand tracking. In Eklundh, J.-O., editor, *Proc. 3rd European Conf. Computer Vision*, 35–46. Springer-Verlag.
- Reid, I. and Murray, D. (1993). Tracking foveated corner clusters using affine structure. In *Proc. 4th Int. Conf. on Computer Vision*, 76–83.
- Reid, I. and Murray, D. (1996). Active tracking of foveated feature clusters using affine structure. *Int. J. Computer Vision*, 18, 1, 41–60.

- Reynard, D., Wildenberg, A., Blake, A., and Marchant, J. (1996). Learning dynamics of complex motions from image sequences. In *Proc. 4th European Conf. Computer Vision*, 357–368, Cambridge, England.
- Rimon, E. and Blake, A. (1996). Caging 2D bodies by one-parameter, two-fingered gripping systems. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1458–1464. IEEE Press.
- Rimon, E. and Burdick, J. (1995a). A configuration space analysis of bodies in contact - I. 1st order mobility. *Mech. Mach. Theory*, 30, 6, 897–912.
- Rimon, E. and Burdick, J. (1995b). A configuration space analysis of bodies in contact - II. 2nd order mobility. *Mech. Mach. Theory*, 30, 6, 913–928.
- Ripley, B. (1987). *Stochastic simulation*. New York: Wiley.
- Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge.
- Ripley, B. and Sutherland, A. (1990). Finding spiral structures in images of galaxies. *Phil. Trans. R. Soc. Lond. A.*, 332, 1627, 477–485.
- Roberts, L. (1965). Machine perception of three - dimensional solids. In Tippet, J., editor, *Optical and Electro-optical Information Processing*. MIT Press.
- Rowe, S. and Blake, A. (1996a). Statistical feature modelling for active contours. In *Proc. 4th European Conf. Computer Vision*, 560–569, Cambridge, England.
- Rowe, S. and Blake, A. (1996b). Statistical mosaics for tracking. *J. Image and Vision Computing*, 14, 549–564.
- Scott, G. (1987). The alternative snake – and other animals. In *Proc. 3rd Alvey Vision Conference*, 341–347.
- Semple, J. and Kneebone, G. (1952). *Algebraic projective geometry*. Oxford University Press.
- Shapiro, L., Zisserman, A., and Brady, J. (1995). 3D motion recovery via affine epipolar geometry. *Int. J. Computer Vision*, 16, 2, 147–182.
- Smith, A. (1996). Blue screen matting. In *Proc. Siggraph*, 259–268. ACM.

Sorenson, H. and Ålspach, D. (1971). Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7, 465–479.

State, A., Hirota, G., Chen, D., Garrett, B., and Livingston, M. (1996). Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In Rushmeier, H., editor, *Proc. Siggraph*, Annual Conference Series, 429–438. ACM SIGGRAPH, Addison Wesley. held in New Orleans, Louisiana, 04-09 August 1996.

Stork, D., Wolff, G., and Levine, E. (1992). Neural network lipreading system for improved speech recognition. In *Proceedings International Joint Conference on Neural Networks*, 2, 289–295.

Storvik, G. (1994). A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16, 10, 976–986.

Strang, G. (1986). *Linear algebra and its applications*. Harcourt Brace.

Strang, G. and Fix, G. (1973). *An analysis of the finite element method*. Prentice-Hall, Englewood Cliffs, USA.

Sullivan, G. (1992). Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, 337, 361–370.

Taylor, M., Blake, A., and Cox, A. (1994). Visually guided grasping in 3D. In *Proc. IEEE Int. Conf. Robotics and Automation*, 761–766.

Terzopoulos, D. (1986). Regularisation of inverse problems involving discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8, 4, 413–424.

Terzopoulos, D. and Fleischer, K. (1988). Deformable models. *The Visual Computer*, 4, 306–331.

Terzopoulos, D. and Metaxas, D. (1991). Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13, 7.

Terzopoulos, D. and Szeliski, R. (1992). Tracking with Kalman snakes. In Blake, A. and Yuille, A., editors, *Active Vision*, 3–20. MIT.

- Terzopoulos, D. and Waters, K. (1990). Analysis of facial images using physical and anatomical models. In *Proc. 3rd Int. Conf. on Computer Vision*, 727–732.
- Terzopoulos, D. and Waters, K. (1993). Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15, 6, 569–579.
- Terzopoulos, D., Witkin, A., and Kass, M. (1988). Constraints on deformable models: recovering 3D shape and nonrigid motion. *J. Artificial Intelligence*, 36, 91–123.
- Thompson, D. and Mundy, J. (1987). Three-dimensional model matching from an unconstrained viewpoint. In *Proc. IEEE Int. Conf. Robotics and Automation*.
- Tomasi, C. and Kanade, T. (1991). Shape and motion from image streams: a factorization method. *Int. J. Computer Vision*, 9, 2, 137–154.
- Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. of Robotics and Automation*, 3, 4, 323–344.
- Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3, 1.
- Uenohara, M. and Kanade, T. (1995). Vision-based object registration for real-time image overlay. In Ayache, N., editor, *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science. Springer-Verlag. ISBN 3-540-59120-6.
- Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13, 10, 992–1006.
- Vaillant, R. (1990). Using occluding contours for 3D object modelling. In Faugeras, O., editor, *Proc. 1st European Conf. Computer Vision*, 454–464. Springer-Verlag.
- Vetter, T. and Poggio, T. (1996). Image synthesis from a single example image. In *Proc. 4th European Conf. Computer Vision*, 652–659, Cambridge, England.
- Waite, J. and Welsh, W. (1990). Head boundary location using snakes. *British Telecom Tech. J.*, 8, 3.

- Waxman, A. and Wohn, S. (1985). Contour evolution, neighbourhood deformation, and global image flow: planar surfaces in motion. *Int. J. Robotics Research*, 4, 95–108.
- Wellner, P. (1993). Interacting with paper on the Digital Desk. *Communications of the ACM*, 36, 7.
- Wildenberg, A. (1997). *Learning and Initialisation for Visual Tracking*. PhD thesis, University of Oxford.
- Williams, L. (1990). Performance-driven facial animation. In *Proc. Siggraph*, 235–242. ACM.
- Witkin, A., Terzopoulos, D., and Kass, M. (1986). Signal matching through scale space. In *5th National Conference on AI*.
- Wloka, M. and Anderson, B. (1995). Resolving occlusion in augmented reality. In *Proc. Symposium on Interactive 3D graphics*, 5–12.
- Yuille, A. (1990). Generalized deformable models, statistical physics, and matching problems. *Neural Computation*, 2, 1–24.
- Yuille, A., Cohen, D., and Hallinan, P. (1989). Feature extraction from faces using deformable templates. *Proc. Conf. Computer Vision and Pattern Recognition*, 104–109.
- Yuille, A. and Grzywacz, N. (1988). A computational theory for the perception of coherent visual motion. *Nature*, 333, 6168, 71–74.
- Yuille, A. and Hallinan, P. (1992). Deformable templates. In Blake, A. and Yuille, A., editors, *Active Vision*, 20–38. MIT.
- Zinkiewicz, O. and Morgan, K. (1983). *Finite elements and approximation*. Wiley, New York.
- Zucker, S., Hummel, R., and Rosenfeld, A. (1977). An application of relaxation labelling to line and curve enhancement. *IEEE trans. comp.*, 26, 4, 394–403.
- Zuniga, O. and Haralick, R. (1983). Corner detection using the facet model. In *Proc. Conf. Computer Vision and Pattern Recognition*, 30–37.

Author Index

- Adjoudani, A., 22
Allen, P.K., 37
Aloimonos, J.Y., 21, 96
Alspach, D.L., 277
Amini, A.A., 22, 36, 139
Anandan, P., 22, 96
Anderson, B.D.O., 277
Anderson, B.G., 23
Arbogast, E., 67
Arika, Y., 254
Astrom, K.J., 211, 278, 298
Ayache, N., 21, 22, 37
Azarbayejani, A., 23
- Bajcsy, R., 21
Baker, H.H., 21, 233
Ballard, D.H., 21, 112, 113
Balter, J., 23
Bandyopadhyay, A., 21
Bar-Shalom, Y., 37, 139, 234, 277, 278
Barnett, S., 67, 156, 281, 293
Barsky, B.A., 67, 284
Bartels, R.H., 67, 284
Bascle, B., 6, 96, 112, 155, 181
Basri, R., 96
Basu, A., 113
Baumberg, A., 22, 113, 184, 254
Beardsley, P., 96
- Beatty, J.C., 67, 284
Belhumeur, P.N., 21, 113
Bellman, R., 139
Bennett, A., 36
Benoit, C., 22
Bergen, J.R., 96, 112
Beymer, D., 184
Binford, T.O., 21
Bing-Hwang, J., 22, 210, 254, 279
Bischofy, B.J., 22
Bishop, C.M., 113, 277
Black, M.J., 22, 37, 96, 184
Blake, A., 7–9, 21, 22, 36, 66, 67, 95,
96, 113, 156, 211, 233, 254,
260, 278, 279, 309
Bobick, A.F., 23, 254
Bodoff, D.A., 22
Bolles, R.C., 37, 233, 277
Bookstein, F.L., 36, 67
Bracewell, R.N., 112
Brady, J.M., 17, 37, 96
Bray, A., 37
Bregler, C., 22, 184
Broida, T.J., 233
Brooke, N.M., 22
Brown, C.M., 21, 37, 67, 112, 113
Bruce, J.W., 184
Bucy, R.S., 277

- Bulthoff, H., 95
 Burdick, J.W., 22
 Burr, D.J., 36
 Burt, P.J., 112
 Buxton, B.F., 21, 95, 138
 Buxton, H., 21, 95

 Campbell, L.W., 23, 254
 Campbell, R., 22
 Canny, J., 21, 28, 112
 Chellappa, R., 233
 Chen, D.T., 23
 Chen, G., 23
 Chow, Y., 184, 278
 Cipolla, R., 23, 36, 37, 66, 67, 95, 96, 309
 Cohen, D.S., 36
 Cohen, I., 22
 Cohen, L.D., 37
 Coombs, D., 37
 Cooper, D.H., 36, 113, 184
 Cootes, T.F., 21, 22, 36, 67, 113, 184
 Cornell, S., 37
 Cox, A., 22
 Cox, I.J., 277
 Craig, J.J., 96
 Craw, I., 36
 Crisman, Jill D., 113
 Curwen, R., 13, 31, 36, 89, 91, 95, 211, 223, 224, 227, 233

 Dalton, B., 14, 22
 Davidson, C., 11, 22, 134
 Demey, S., 96
 Dempster, A., 254
 Deriche, R., 37, 96, 112

 de Boor, C., 67
 Dickmanns, E.D., 22, 37, 233
 di Bernardo, E., 22
 Dodd, B., 22
 Dreschler, L., 22
 Dreyfus, S., 139
 Du, F., 37
 Duda, R.O., 277
 Duncan, J., 22
 Durrant-Whyte, H., 22, 37, 277

 Elschlager, R.A., 36, 95
 Enkelmann, W., 95
 Essa, I., 22
 Ettinger, G.J., 23

 Farrokhnia, F., 112
 Faugeras, O., 21, 37, 96, 303
 Faux, I.D., 67, 184, 284
 Faverjon, B., 21, 22, 67
 Feiner, S.K., 22, 67, 96, 112, 113, 284, 303
 Ferri, F., 22, 113
 Ferrier, N.J., 7–9, 22
 Finn, E.K., 22
 Fischler, M.A., 36, 37, 95, 277
 Fitzgibbon, A.W., 104
 Fix, G.J., 36
 Flannery, B.P., 67, 137
 Fleischer, K., 37
 Foley, J.D., 22, 67, 96, 112, 113, 284, 303
 Forsyth, D., 67
 Fortmann, T.E., 37, 139, 234, 277, 278
 Freeman, W.T., 156
 Frisby, J., 21

- Garrett, B., 23
 Gee, A., 23, 37
 Gelb, A., 37, 211, 234, 252, 277, 278, 298
 Gelfand, A.E., 278
 Geman, D., 278
 Geman, S., 278
 Gennery, D.B., 36, 233
 Ghali, N.M., 23, 254
 Giblin, P., 96, 184, 309
 Golub, G.H., 184, 281
 Goncalves, L., 22
 Gonzales, R.C., 67, 137
 Goodwin, C.G., 252
 Gordon, N., 278
 Graefe, V., 22, 37, 233
 Graham, J., 36, 113, 184
 Grenander, U., 21, 184, 278
 Grimson, W.E.L., 21, 23, 37, 137
 Grzywacz, N.M., 210
 Guillemaud, R., 17

 Hager, G., 37, 113
 Hainsworth, T.J., 23, 254
 Hallam, J., 37
 Hallinan, P., 33, 36, 95
 Hampel, F.R., 139
 Hanna, K.J., 96
 Haralick, R.M., 21, 112
 Harris, C., 36, 37, 95, 96, 112, 137, 234
 Hart, P.E., 277
 Haslam, J., 22
 Hebert, M., 21
 Heeger, D., 95
 Heller, A.J., 21
 Herlin, I., 22

 Hespanha, J.P., 113
 Heuring, J.J., 23, 37
 Hildreth, E.C., 21, 138
 Hill, A., 22
 Hingorani, R., 96, 112
 Hinton, G.E., 36
 Hirota, G., 23
 Hogg, D., 22, 113, 184, 254
 Horn, B.K.P., 95, 137, 138
 Horowitz, B., 23
 Howes, M., 23, 254
 Huang, X.D., 254
 Hughes, J.F., 22, 67, 96, 112, 113, 284, 303
 Hummel, R.A., 21
 Huttenlocher, D.P., 37

 Ikeuchi, K., 137
 Inoue, H., 137
 Isard, M.A., 254, 278, 279
 Ivins, J., 96, 112

 Jack, M.A., 254
 Jacob, G., 16, 197
 Jacobs, O.L.R., 139, 277
 Jain, A.K., 112
 Jepson, A.D., 184
 Juste, F., 22, 113

 Kanade, T., 21, 23, 37, 68, 95, 96, 112, 156, 233
 Kass, M., 21, 36, 37, 68, 95, 112, 139
 Kaucic, R., 14, 22, 106, 108, 176, 180
 Keenan, D.M., 184, 278
 Kendall, M.G., 252, 300
 Khosla, P.K., 37
 Kikinis, R., 23

- Kitagawa, G., 278
 Kitchen, L., 112
 Kneebone, G.T., 96
 Koch, C., 137
 Koenderink, J.J., 96, 309
 Koller, D., 22, 95, 113
 Konig, Y., 22
 Kreysig, E., 67
 Kriegman, D.J., 67, 113
 Kutulakos, K.N., 23

 Laird, M., 254
 Landau, L.D., 211
 Lanitis, A., 21, 22, 36, 113, 184
 Lawn, J.M., 96
 Levin, D., 23
 Levine, E., 22
 Lifshitz, E.M., 211
 Little, J., 95
 Livingston, M., 23
 Ljung, L., 252, 254
 Lowe, D.G., 36, 95, 137
 Lozano-Perez, T., 21, 23
 Lucas, B.D., 112

 McCormick, J., 260, 278
 Malik, J., 21, 22, 95, 113
 Marais, P., 17
 Marchant, J., 10, 22, 254
 Mardia, K.V., 23, 254
 Marimont, D.H., 233
 Marinos, C., 95, 156
 Marr, D., 21
 Matthies, L.H., 37, 233
 Mayhew, J., 21, 37
 McLauchlan, P.F., 22, 37

 Medioni, G., 36, 67, 95
 Menet, S., 36, 67, 95
 Metaxas, D., 37, 95
 Mizoguchi, H., 137
 Mohr, R., 67
 Montanari, U., 21
 Montgomery, A.A., 22
 Moore, J.B., 277
 Morgan, K., 36
 Morris, J., 6, 13
 Mumford, D., 21
 Mundy, J.L., 21, 22, 36, 67, 96
 Murase, H., 95
 Murray, D.W., 12, 22, 23, 37, 96, 112, 113, 138

 Nagel, H.H., 22, 95
 Nayar, S.K., 95
 Noble, J.A., 16, 112, 197
 Noh, J.J., 37
 North, B., 6, 20, 254

 O’Gorman, F., 21
 Ohta, Y., 21, 68
 Omohundro, S.M., 22, 184
 Owen, R., 22

 Papanikolopoulos, N., 37
 Papoulis, A., 67, 184, 211, 294
 Pardey, J., 23
 Peleg, S., 112
 Pelizzari, C., 23
 Pentland, A., 22, 23, 184
 Perona, P., 21, 22
 Petajan, E.D., 22
 Petitjean, S., 67
 Plá, F., 22, 113

- Poggio, T., 95, 137, 184
Pollard, S.B., 21
Ponce, J., 22, 67
Porrill, J., 96, 112
Pratt, M.J., 67, 184, 284
Press, W.H., 67, 137
- Rabiner, L., 22, 210, 254, 279
Ramer, E.U., 21
Rao, B., 22, 37, 277
Rao, C.R., 184, 252
Rehg, J., 37, 95, 96
Reid, I.D., 22, 37, 96, 112
Revow, M.D., 36
Reynard, D., 10, 22, 254
Rimon, E., 22
Ripley, B.D., 184, 278
Roberts, L.G., 21
Roberts, S., 23
Ronchetti, E.M., 139
Rosenfeld, A., 21, 112
Rousseeuw, P.J., 139
Rowe, S.M., 7–9, 12, 22, 113
Rubin, D., 254
Rucklidge, W.J., 37
- Saint-Marc, P., 36, 67, 95
Sakuma, M., 17
Salmond, D., 278
Schunk, B.G., 95, 137, 138
Scott, G.L., 36
Semple, J.G., 96
Shapiro, L., 96
Sharkey, P.M., 22, 37
Sheehy, N., 23, 254
Sheen, J.A., 22, 37, 277
- Sin, K.S., 252
Smith, A.F.M., 278
Smith, A.R., 22
Smith, S., 110
Soong, J., 37
Sorenson, H.W., 277
Stahel, W.A., 139
Staib, L., 22
Starner, T., 23
State, A., 23
Stennett, C., 137
Stork, D., 22
Storvik, G., 184, 278
Strang, G., 36, 96
Stuart, A., 252, 300
Sullivan, G.D., 22, 36, 112
Sutherland, A.L., 184, 278
Szeliski, R., 36, 37, 211, 233
- Tan, K., 23
Tarassenko, L., 23
Taylor, C.J., 21, 22, 36, 67, 113, 184
Taylor, M., 22
Tehrani, S., 36, 139
Tenenbaum, J.B., 156
Terzopoulos, D., 21, 36, 37, 68, 95,
112, 137, 139, 211
Teukolsky, S.A., 67, 137
Thompson, D.W., 36
Timcenko, A., 37
Tomasi, C., 156
Torre, V., 137
Tsai, R.Y., 96
Turk, M., 184
- Uenohara, M., 23

- Ullman, S., 96
 Ursella, E., 22
- Vaillant, R., 96, 309
 Valliano, J.R., 23
 van Dam, A., 22, 67, 96, 112, 113, 284, 303
 van Doorn, A.J., 96
 van Loan, C.F., 184, 281
 Vetter, T., 184
 Vetterling, W.T., 67, 137
 Vicens, M., 22, 113
- Waite, J.B., 22
 Waters, K., 21, 36, 95
 Waxman, A.M., 138
 Weber, J., 22, 95, 113
 Weiss, I., 21
 Weiss, R., 96, 309
 Wellner, P., 23
 Wells, W.M., 23
 Welsh, W.J., 22
 Weymouth, T.E., 36, 139
 White, S.J., 23
 Wildenberg, A.P., 10, 22, 95, 254
 Williams, C.K.I., 36
 Williams, L., 21
 Wilson, A.D., 23, 254
 Wintz, P., 67, 137
 Witkin, A., 21, 36, 37, 68, 95, 112, 139
 Wittenmark, B., 211, 298
 Wloka, M.M., 23
 Wohn, S., 138
 Wolff, G., 22
- Yacoob, Y., 22, 37, 96
 Yasumoto, Y., 67
- Yoshimi, B., 37
 Yuille, A.L., 21, 33, 36, 95, 210
- Zheng, Y., 37
 Zinkiewicz, O.C., 36
 Zisserman, A., 17, 22, 67, 95, 96, 211, 233
 Zucker, S., 21
 Zuniga, O.A., 112

Index

Italic entries denote references in the Bibliographic notes.

- accuracy
 - of learning, 241, 300
 - of pose recovery, 149, 153
- active vision, 2, *21*
- actor-driven animation, 5, *21*
- adaptive search, 221
- affine
 - derivation of planar, 78
 - moments of planar, 80
 - planar, 72, 76, *95*, 141, 195
 - three-dimensional, 88, 149, 154
- aggregated measurement, 126, 127, 171, 214, 217, 257
- aggregation of training sets, 243
- algorithm
 - CONDENSATION, 255, 263, *278*
 - alternative curve-fitting, 131
 - alternative Kalman filter, 220
 - B-spline span matrix, 291
 - construction of posterior, 171
 - curve-fitting, 127, 130, *138*
 - curve-fitting with corners, 132
 - factored sampling, 257
 - Fisher discriminant, 107
 - Kalman filter, 218, 220
 - learning motion, *252*
 - learning motion in 1D, 236, 238
 - learning motion in shape-space, 242, 244, 245
 - planar pose recovery, 142, 143
 - principal components analysis, 179, 183
 - rigid/non-rigid decomposition, 154, 155
 - three-dimensional pose recovery, 151, 153, *156*
- ambiguity in pose recovery, 143, 149
- animation, actor-driven, 5, *21*
- aperiodic B-spline, 45, 289, 291
- aperture problem, *137*
- appendage of non-rigid body, 94, 305
- approximation, least-squares, 51, 130, *139*
- arclength, 64, 126
- area of B-spline, 63, 65, 66, 80, 293
- ARP, *see* auto-regressive process
- articulated bodies, 93, *96*, 273, 303
- assimilation of measurements, 126, 215, 216, 220
- augmented shape-space, 181
- auto-correlation coefficients, 238, 244, 301

- auto-regressive process
 - first order, 193, 297
 - second order, 200, 225, 238, 261, 263, 299
- average curve displacement, *see also* norm
 - dynamic, 195, 208, 225, 227
 - static, 60, 62, 79, 163, 165
- B-spline, 41, 284
 - aperiodic, 45, 289, 291
 - area, 63, 65, 66, 80, 293
 - basis, 42, 284
 - construction, 42, 284
 - corner, 54, 287
 - degree, 41
 - discontinuity, 46, 54, 287
 - function, 42, 67, 284
 - hinge, 54, 287
 - implementation, 287
 - inertia, 80
 - metric matrix, 50, 292
 - norm, 47
 - normalisation, 285
 - order, 41
 - periodic, 45, 53, 287
 - quadratic, 41
 - regular, 42
 - smoothness, 47, 285, 287
 - vertex, 54, 287
 - weights, 43, 53
- background subtraction, 110, 113
- backward difference, 298
- basis
 - B-spline, 42, 284
 - PCA, 178
 - planar affine, 73, 78
 - shape-space, 74
 - three-dimensional affine, 89
- Bayes' rule, 160, 164, 265, 295
- biometrics, 11
- blue-screening, 17, 22, 277
- Boltzmann distribution, 172
- bootstrap filtering algorithm, 278
- bootstrapping, 251, 269, 274
- breakpoint, 41, 46, 287
- Bresenham line algorithm, 112
- Brownian motion, 188, 297
- calculus of variations, 121
- calibration, camera, 7, 84, 96
- car tracking, 7, 196, 208
- cardiac analysis, 15, 90
- centroid, 63, 65, 75, 80, 151
- chain
 - kinematic, 95, 306
 - Markov, 188, 210
- classification of signals, 23
- closed-loop steady state, 219
- clutter resistance, *see also* background subtraction
 - in target tracking, 278
- problem, 18
- using colour, 106
- using dynamics, 134
- using shape-space, 72
- using validation gate, 224
- using CONDENSATION, 255, 262
- coherence, 239
- colour, 8, 82, 104, 113, 232
- computational complexity, 67, 132, 220
- CONDENSATION, 4, 255, 259, 278

- conditional distribution, 294
- confidence region, 161, 163, 184, 188, 296
- configuration vector, *see* shape-vector
- constant velocity, 196, 207, 211, 219, 221
- construction
 - of B-spline, 42, 284
 - of dynamical model, 207, 243
 - of shape-space, 69, 90, 153, 179, 303
- continuous-time dynamics, 197, 205, 211, 230, 241, 247, 255, 297
- control point, 53
- control polygon, 54
- control vector, 57
- conversion between continuous- and discrete-time, 298
- convolution, 101, 112
- coordinate vector, *see* control vector
- corner
 - image feature, 101, 112, 131
 - in B-spline, 54, 287
- correlation, 105, 112
- correspondence, point, 37
- cost, computational, 67, 132, 220
- covariance
 - definition, 295
 - dynamic, 194, 205, 298
 - for validation gate, 220
 - form of curve-fitting algorithm, 131
 - static, 160, 176
 - static prior for dynamic model, 247
- critical damping, 206
- curve approximation
 - as projection, 52, 67
 - from measurements, 115, 127, 131, 132, 137
 - probabilistic, 159, 184, 257, 278
 - with outliers, 174
- curve norm, 58, 67
- damping, 200, 206, 225, 240
 - critical, 206
- decay, 198, 240, 248, 297
- decomposition
 - for pose recovery, 143, 145, 149
 - of rigid and non-rigid motion, 153
 - singular value, 149, 153, 156, 284
- default dynamical model, 205
- deformable template, 32, 36, 95, 136
- deformation space, 190, 196, 207, 229
- degeneracy of pose recovery, *see* ambiguity in pose recovery
- degree of B-spline, 41
- density
 - conditional, 294
 - function, 294
 - Gaussian, 160, 184, 296
 - multi-modal, 255, 266, 267
 - non-Gaussian, 255, 277
 - observation, 160, 164, 169, 171, 215, 255, 258, 262, 295
 - posterior, 159, 164, 167, 171, 215, 258, 295
 - prior, 160, 184, 215, 257, 295
 - process, 192, 193, 215, 297
 - propagation, 255, 259
 - spectral, 200, 298
 - state, 256, 262

- detection
 - edge, 26, 27, 101, 112
 - feature, 21, 26, 97, 112
- deterministic model, 35
- DFT, *see* Discrete Fourier Transform
- diagonalisation of square matrix, 284
- difference, image, 112, 113
- differential equation, *see* stochastic differential equation
- diffusion, 255
- digital desk, 18, 23, 231
- discontinuity in B-spline, 46, 54, 287
- Discrete Fourier Transform, 109, 198
- discrete-time dynamics, 190, 205, 211, 298
- discriminant, Fisher, 105, 107, 113
- distraction, *see* clutter resistance
- distribution, *see also* density
 - Boltzmann, 172
 - definition, 294
 - dynamical prior, 186, 190
 - Gaussian, 295
 - learning prior, 174
 - of curve shape, 159, 160
 - probabilistic framework for image interpretation, 3
 - sampling from, 163
 - steady-state
 - first-order, 193, 195, 196
 - second-order, 205–207, 230, 247, 254, 269, 301
- drift of random walk, 189, 195
- dynamic contours, 34, 36, 213
- dynamic programming, 139
- dynamical model, 185, 235, 297
- edge feature, 26, 27, 101, 112
- efficiency, *see* computational complexity
- eigenvalue, eigenvector, 283
 - for pose recovery, 142
 - of dynamical prediction matrix, 203, 248, 297
 - of shape covariance, 161, 177, 184
- elasticity of snake, 28
- ellipse, covariance, 161, 194
- EM, *see* expectation–maximisation
- energy
 - of motion model, 201, 211
 - of snake, 28, 97
- envelope of AR process, 190, 205, 247
- equilibrium equation for snake, 28
- estimation
 - curve, *see* curve approximation
 - least-squares, 51, 130, 139
 - of motion model, 235
- Euclidean
 - distance measure, 58
 - norm, 59, 79, 177, 282
 - similarity, 70, 75, 80, 118, 303
- Euler angle, 143
- evolution of state density, 213, 255, 278
- expectation of random variable, 294
- expectation–maximisation, 240, 254
- extended Kalman filter, 277
- factored sampling, 257, 278
- factorisation
 - image-stream, 153, 156
 - of rigid and non-rigid motion, 153
- feature curve, 97, 116, 137, 164, 169

- feature detection, *21, 26, 97, 112*
- feature, image, *see* image feature
- field of view, *85–87, 141, 147*
- first-order AR process, *193, 297*
- first-order dynamical model, *188, 297*
 - limitations, *196*
- Fisher discriminant, *105, 107, 113*
- Fisher information measure, *300*
- fitting, *see* curve approximation
- flow, optical, *21, 37, 81, 95*
- focal length, *83, 96, 143*
- Fokker-Planck equation, *255, 278*
- frequency of periodic motion, *198, 206, 225, 237*
- function, B-spline, *42, 67, 284*
- gain, Kalman, *216, 234, 239*
- Gaussian
 - density, *160, 184, 295*
 - distribution, *295*
 - Markov process, *188*
 - smoothing, *101*
- gesture recognition, *23, 254*
- Gibbs sampler, *278*
- grasp, *8, 22*
- harmonic motion, *200, 206, 207, 209, 230, 248*
- Hidden Markov Model, *22, 279*
- hinge in B-spline, *54, 287*
- hinged appendage, *94, 303*
- homogeneous coordinates, *22, 303*
- hue, *105, 113*
- image difference, *112, 113*
- image feature, *26, 97*
 - corner fitting, *131*
 - for CONDENSATION, *265*
 - probabilistic, *169, 171*
- image processing, *97, 137, 278*
- image sequence, *188, 213, 238, 259*
- image-stream factorisation, *153, 156*
- implementation of B-splines, *287*
- importance sampling, *278*
- impulse response, *198, 200, 247*
- independence of measurements, *170, 171, 266*
- inertia
 - of B-spline, *80*
 - of snake, *34*
- information, statistical, *see* statistical information
- initialisation, *63, 81, 95, 217, 220, 268*
- inner product, *see* norm
- innovation, *124, 130, 173*
- interpolated image sampling, *99, 112*
- invariance, parameterisation, *59, 64, 65, 67, 80, 121*
- invariant norm, *122*
- invariant subspace, *117, 169, 173*
- isotropic prior, *161*
- iterative learning, *see* bootstrapping
- iterative least-squares estimation, *51, 130, 139*
- JPDAF, *277*
- Kalman filter, *37, 213, 220, 233, 255*
 - Extended, *277*
 - for recursive fitting, *131*
 - for snakes, *35*
- Kalman gain, *216, 234, 239*
- key-frames, *90, 153, 177*

- kinematic model, 94, 96, 303
- knot, 42, 284
 - multiple, 46, 54, 287
- Kronecker product, 282
- L_2 norm, 47, 59, 79
- L_2 norm PCA, 178, 183, 184
- Lagrangian dynamics, 36, 203, 211
- learning
 - dynamical model, 235
 - Fisher discriminant, 105
 - prior for shape, 174
- least-squares estimation, 51, 130, 139
- likelihood, 160, 295
 - maximum, 175, 237, 252, 295, 300
- limiting covariance, 193
- linear discriminant, 105, 107, 113
- linear requirement for shape-space, 69
- linear scanning, 99
- local invariance to parameterisation, 121
- low-level vision, 21
- magnitude of a vector, 282
- Mahalanobis metric, 184
- manual selection of dynamical
 - parameters, 205, 225, 231
- Markov chain, 188, 210
- Markov Chain Monte-Carlo
 - simulation, 278
- Markov process, 188, 193, 210
- Markov Random Field, 184
- mask, for feature detection, 26, 112
- matching, *see* feature detection
- matrix operations, 281
- maximum a posteriori estimate, 167, 295
- maximum likelihood estimate, 175, 237, 252, 295, 300
- MCMC, *see* Markov Chain Monte Carlo simulation
- mean estimation, 242, 244, 254, 258
- mean of random variable, 294
- mean-square displacement, 161, 169, 172
- mean-state equation, 193, 205, 297
- measurement assimilation, 126, 215, 216, 220
- measurement error, 126, 264
- measurement history, 213, 260
- metric matrix
 - B-spline, 50, 292
 - curve, 59
 - shape-space, 79
- metric, Mahalanobis, 184
- mixture of Gaussians filter, 277
- MLE, *see* Maximum Likelihood Estimation
- modal analysis, 248
- model
 - constant velocity, 207
 - examples of dynamics, 208, 231
 - first-order dynamical, 188, 297
 - learning dynamics, 235
 - learning shape prior, 174, 184
 - limitations of first-order, 196
 - observation for CONDENSATION, 262, 278
 - probabilistic feature, 169
 - probabilistic shape, 159, 184
 - second-order dynamical, 200, 201, 297

- setting default dynamics, 205, 225, 231
 - shape, 69
- modular learning, 243
- moments
 - area, 63, 65, 66, 80, 293
 - estimation for CONDENSATION, 262
 - for initialisation, 81, 95
 - in shape-space, 79
 - invariant, 65, 80
 - of curves, 63
- morphological filtering, 113
- MRF, *see* Markov Random Field
- multi-modal density, 255, 266, 267
- multi-variate Gaussian distribution, 295
- multi-variate learning algorithm, 242
- multiple control point, 54
- multiple knot, 46, 54, 287
- multiple-hypothesis filtering, 255, 277
- nodal variable, 30
- noise
 - in AR process, 198, 201, 207, 211, 297
 - in feature detection, 101, 170
 - resistance to, 117, 131
- non-Gaussian density, 255, 277
- non-linear filtering, 256, 277
- non-linear shape models, 94, 95, 234
- non-periodic B-spline, 45, 289, 291
- non-rigid motion, 90, 153, 176
- norm
 - L_2 , 47, 59, 79
 - B-spline, 47
 - curve, 58, 67
 - Euclidean, 59, 79, 177, 282
 - parameterisation invariant, 122
 - shape-space, 79
 - weighted, 125
- normal displacement, 63, 120, 122, 137, 162, 169
- normal form for SDE, 299
- normal variable, *see* Gaussian distribution
- normal vector search-lines, 98, 113, 122, 161, 266, 278, 292
- normalisation
 - B-spline, 285
 - probability distributions, 294
 - vector, 282
- observation
 - aggregated, 126, 127, 171, 214, 217, 257
 - density, 160, 164, 169, 171, 215, 255, 258, 295
 - error, 165, 225, 230, 240, 254
 - history, 213, 260
 - model, 170, 262, 278
- occlusion, 88, 224, 264
- open-loop steady state, 219
- operator, image-feature, 26, 101, 112
- optical flow, 21, 37, 81, 95
- order of B-spline, 41
- orthogonal matrix, 283
- orthographic projection, 87, 141
- oscillator, harmonic, 206, 209, 299
- oscillatory motion, 200, 206, 247, 299
- outlier, 130, 139, 172, 174
- parallax, 149

- parameterisation invariance, 59, 64, 65, 67, 80, 121
- parameters
 - learning, 235, 252, 300
 - setting manually, 205, 225, 231
- paraperspective, *see* weak perspective
- partition of motion across
 - shape-space, 207, 230
- partition of shape-space, 190, 196
- pattern theory, 2, 21
- PCA, *see* Principal Components Analysis
- PDAF, 277
- PDM, *see* Point Distribution Model
- performance of tracker, 220, 231, 250
- periodic B-spline, 45, 53, 287
- periodic motion, 196, 206, 239
- perspective, 72, 81, 96
 - weak, 81, 85, 96, 147
- phase, signal, 203, 238, 249
- physical realisability of dynamical
 - model, 201, 211
- planar-affine, 72, 76, 95, 141, 195
- point correspondence, 37
- Point Distribution Model, 177, 184
- Poisson process, 264
- polar angles in pose recovery, 148
- pole of dynamical model, 202, 297
- polygon, control, 54
- pose recovery, 141
 - ambiguity, 143, 149
 - of planar object, 141–143, 156
 - three-dimensional, 149, 151, 153, 156
 - weak perspective, 147, 148
- posterior, 159, 164, 167, 171, 215, 258, 295
- power spectrum, 198, 200, 201, 211, 237, 298, 300
- Principal Components Analysis, 176, 179, 183, 184
- prior, 160, 184, 215, 257, 295
- probabilistic model
 - dynamical, 185
 - observation, 169, 264
 - shape, 159, 184
- projection
 - onto shape-space, 116, 118, 161, 207
 - orthographic, 87, 141
 - perspective, 72, 81, 96
 - weak perspective, 81, 85, 96, 147
- propagation of state density, 205, 214, 255
- pseudo-inverse, 67, 79, 118, 282
- quadratic B-spline, 41
- radius, spectral, 284
- random sampling, 37, 163, 243, 255, 278
- random variable, 294
- random walk, 188, 278
- rank of a matrix, 282
- RANSAC, 277
- real-time tracking, 5, 31, 112, 139, 265, *see also* computational complexity
- realisability of physical model, 201, 211
- recognition by synthesis, 3, 257

- recovery, pose, *see* pose recovery
- recursive least-squares, *see*
 - least-squares estimation
- region-based image processing, 95, 101, 109, 112
- regular B-spline, 42
- regularised matching, 115, 118, 137, 160
- reparameterisation invariance, *see*
 - parameterisation invariance
- residual PCA, 178
- resonant peak, 199, 200, 211, 238
- Riccati equation, 194, 297
- ridge feature, 27, 101, 112
- rigid motion, 70, 153
- RMS, *see* root-mean-square
- robust estimator, 139, *see* validation
 - gate
- robustness to clutter, *see* clutter
 - resistance
- robustness to illumination, 109, 112
- root-mean-square, 49
 - displacement, *see* average curve
 - displacement
- rotation matrix, 283
- sample
 - covariance, 176, 254
 - mean, 175
 - path, 193
 - point, 98, 123
 - set, 257
- sampling
 - interpolated image, 99
 - random curve, 164, 257
- scalar product, 281
- scan, linear, 99
- SDE, *see* stochastic differential
 - equations
- search region, 97, 131, 137, 162, 220, 221, 264
- second-order dynamical model, 200, 201, 203, 238, 247, 299
- segment, search, 97, 131, 137, 162, 220, 221, 264
- segmentation, colour, 8, 81, 113
- selectivity for motion, 250
- sequence, image, 188, 213, 238, 259
- setting parameters, 205, 225, 231
- shape-space, 69, 74, 95
 - constructive, 69, 90, 153, 179, 303
 - examples, 75, 76
 - justification of linear, 93, 96
 - key-frame, 90, 153, 177
 - moments, 79
 - norm, 79
 - partition, 190, 196
 - PCA, 176, 179, 183, 184
 - planar affine, 72, 76, 95, 141, 195
 - projection, 116, 118, 161, 207
 - three-dimensional affine, 88, 149, 154
- shape-vector, 69, 74
- signal phase, 203, 238, 249
- signal power, *see* power spectrum
- signal processing, 67, 100, 112
- silhouette curve shape-spaces, 309
- similarity, *see* Euclidean similarity
- simulated annealing, 278
- simulation, 37, 163, 243, 255, 278
- singular matrix, *see* numerical
 - stability

- Singular Value Decomposition, 149, 153, 156, 284
- singularity of planar pose recovery, 149
- smoothing of tracked sequence, 278
- smoothing, Gaussian, 101
- smoothness
 - of B-spline, 47, 285, 287
 - of snake curve, 28, 30
- snake, 2, 21, 26, 36, 233
 - elasticity, 28
 - energy, 28, 97
 - equilibrium equation, 28
 - inertia, 34
 - Kalman filter, 35
 - smoothness, 28, 30
 - stability, 30
 - stiffness, 28
 - viscosity, 34
- space, *see* shape-space
- span, 41, 287
 - matrix, 46, 288, 291
- spatial variance, 161, 167, 206
- spectral
 - analysis, 198, 211, 237, 298
 - density, 200, 298
 - radius, 284
- spectrum, power, 198, 200, 201, 211, 237, 298, 300
- spline, *see* B-spline
- spline-vector, *see* control vector
- square-root, matrix, 284
- stability
 - numerical, 78, 134
 - of dynamical model, 193, 203, 206, 284, 298
 - of snake, 30
- state density, 256, 262
- state-vector, 204
- static curve-fitting, *see* curve approximation
- statistical information, 296
 - in dynamical approximation, 185, 214, 216, 234
 - in static approximation, 126, 131, 160, 167, 171
- steady-state distribution
 - first-order, 193, 195, 196
 - second-order, 205–207, 230, 247, 254, 269, 301
- stiffness of snake, 28
- stochastic differential equations, 206, 211, 297
- stochastic model, *see* auto-regressive process; probabilistic shape model; second-order dynamical model; first-order dynamical model
- subspace
 - invariant, 117, 169, 173
 - of shape-space, 190, 196
- subtraction, background, 110, 113
- SVD, *see* Singular Value Decomposition
- synthesis
 - of model, *see* parameters, setting manually
 - of sequence, *see* random sampling
- telescopic appendage, 94, 307
- template
 - deformable, 32, 36, 95, 136
 - intensity, *see* correlation matching

- of default shape, 32, 70, 72, 74–76, 89, 120, 159
- threshold
 - for outliers, *see* validation gate
 - in feature detection, 26, 101, 111
- time to collision, 66
- time-course, *see* impulse response
- trace of matrix, 283
- traffic monitoring, 6, 22, 95, 136, 208
- training, *see* learning
- transformation between continuous- and discrete-time dynamics, 298
- translation
 - invariance to regularisation, 134, 168
 - unconstrained motion, 190, 196, 207, 229
- transpose of matrix, 282
- trough, 27, 101, 112
- tuning, *see* parameters, learning;
 - parameters, setting manually
- uncertainty, *see* probabilistic model;
 - noise
- unconstrained
 - Brownian motion, 188, 194
 - with drift, 195
 - shape prior, 173
 - translation, 190, 196, 207, 229
- underconstrained
 - measurement process, 134, 137, 138
 - shape-space, 72, 94
- uniform line-reading, 99
- uniform measurement sampling, 126
- uniform shape prior, 161, 167
- univariate learning, 236
- unstable, *see* stable; underconstrained
- validation gate, 130, 139, 172, 220, 233
- validity
 - of orthographic projection, 87
 - of weak perspective, 85
- valley feature, 27, 101, 112
- variability, *see* covariance; spatial
 - variance
- variance
 - of random variable, 294
 - spatial, 161, 167, 206
- vector
 - colour, 104
 - control, 57
 - Fisher, *see* Fisher discriminant
 - magnitude, 282
 - normal, *see* normal vector
 - search-lines
 - observation, 126, 127, 171, 214, 217, 257
 - operations, 281
 - product, 281
 - state, 204
 - unit, 282
- velocity initialisation, 219
- velocity, constant, 196, 207, 211, 219, 221
- vertex in B-spline, 54, 287
- virtual orthographic image, 147
- viscosity of snake, 34
- visibility, *see* occlusion
- weak perspective, 81, 85, 96, 147

weight matrix, 117, *see* regularised
 matching
weighted norm, 125
weighted sample, 257, 260
weighting, information, *see* statistical
 information
weights, B-spline, 43, 53
white noise, 203, 297
Wiener process, 203
writing, dynamics of, 243

Yule-Walker equations, 252

zero of dynamical model, 202