Chapter 1

Introduction

Psychologists of vision have delighted in various demonstrations in which prior knowledge helps with interpreting an image. Sometimes the effects are dramatic, to the point that the viewer can make no sense of the image at all until, when cued with a single word, the object pops out of the image. This idea of "priming" with prior knowledge is illustrated (light-heartedly) in figure 1.1. Priming in that example is

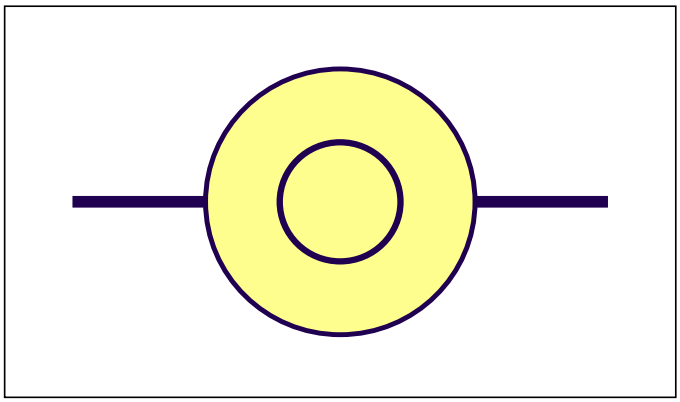


Figure 1.1: Priming with prior knowledge. *If you have never seen it before this figure probably means little at first sight. Now look for a cyclist in a Mexican hat.*

rather "high-level," calling on some intricate and diverse common-sense knowledge concerning wheels, hats and so on. The aim of this book is to look at how prior

knowledge can be applied in machine vision at the lower level of shapes and outlines.

The attraction of using prior knowledge in machine vision is simply that it is so hard to make progress without it, as a decade or more of research around the 1970s showed. There was considerable success in converting images into something like line drawings without resorting to any but the most general prior knowledge about smoothness and continuity. That led to the problem of "grouping" together the lines belonging to each object which is difficult in principle, and very demanding of computing power. One effective escape from this bind has been to design vision processes in a more goal-directed fashion and this is part of the philosophy of the notably successful "Active Vision" paradigm of the 1980s. Consider the task of examining visually the field of view immediately in front of a driverless vehicle, in order to steer automatically along the road. If the nature of the task is taken into account from the outset, it is quite unnecessary to examine an entire image; it is sufficient to focus on the expected appearance and position of the road edge at successive times. Deviations of actual from expected position can be treated as an error signal to control steering. This has two great advantages. First there is no need to organise or group features in the image; the relevant area of the image is simply tested directly against its expected appearance. Secondly, the fact that analysis can be restricted to a relatively narrow "region of interest" (around the road edge) eases the computational load. Active Vision, then, uses task-related prior knowledge to simplify and focus the processing that is applied to each image.

This book is concerned with the application of prior knowledge of a particular kind, namely geometrical knowledge. The aim is to strengthen the visual interpretation of shape via the stabilising influence of prior expectations of the shapes that are likely to be seen. There have been many influences in the development of this approach and two in particular are outstanding. First is the seminal work in 1987 of M. Kass, A. Witkin and D. Terzopoulos on "snakes" which represented a fundamentally new approach to visual analysis of shape. A snake is an elastic contour which is fitted to features detected in an image. The nature of its elastic energy draws it more or less strongly to certain preferred configurations, representing prior information about shape which is to be balanced with evidence from an image. If also inertia is attributed to a snake it acquires dynamic behaviour which can be used to apply prior knowledge of motion, not just of shape. Snakes are described in detail in the next chapter. The second outstanding influence is "Pattern Theory" founded by U. Grenander in the 70s and 80s and a popular basis for image interpretation in the statistical community. It puts the treatment of prior knowledge about shape into a

probabilistic context by regarding any shape as the result of applying some distortion to an ideal prototype shape. The nature and extent of the distortion is governed by an appropriate probability distribution which then effectively defines the range of likely shapes.

Defining a prior distribution for shape is only part of the problem. The complete image interpretation task is to modify the prior distribution to take account of image features, arriving at a "posterior" distribution for what shape is actually likely to be present in a particular image. Mechanisms for fusing a prior distribution with "observations" are of crucial importance. Suffice it to say here that a key idea of pattern theory is "recognition by synthesis," in which predictions of likely shapes, based on the prior distribution, are tested against a particular image. Any discrepancy between what is predicted and what is actually observed can be used as an error signal, to correct the estimated shape. Fusion mechanisms of this general type exist in the snake, in the ubiquitous "Kalman filter" described in the next chapter, and in other more general forms described later in the book.

1.1 Organisation of the book

The organisation of material in the book is as follows. This chapter concludes by illustrating a range of applications and the next introduces active contour models. The book is then divided into two parts. Part I deals with the fundamentals of representing curves geometrically using splines, including basic machinery for least-squares approximation of spline functions, an essential topic not normally dealt with in graphics texts. Chapter 4 lays out a design methodology for linear, image-based, parametric models of shape, an important tool in applying shape constraints. Then algorithms for image processing and fitting splines to image features are introduced, leading to practical deformable templates in chapter 6. At this stage, a tool-set has been amassed sufficient for fitting curves to individual images, under a whole spectrum of prior assumptions, ranging from the least constrained snake to a two-dimensional rigid template. The treatment of part I aims to be thorough and complete, accessible by readers who are not necessarily familiar with the techniques of computer vision, given just a reasonable background in computing and vector algebra. (Appendix A reviews the necessary background in vectors and matrices, and gives some additional implementation details on spline curves.)

Part II introduces two new themes: models of motion and deformation, and prob-

abilistic treatment of shape and motion. It begins (chapter 8) by reinterpreting the deformable templates of part I, in probabilistic terms. This is extended to dynamical models in chapter 9, as a preparation for fully probabilistic dynamical contour tracking, by Kalman filter, in chapter 10. By this stage, there are numerous parameters to be chosen to build a competent tracker and clear design guidelines are given on setting those parameters and on their intuitive physical interpretations. The most effective dynamical models derive, however, from learning procedures, as described in chapter 11, in which tracking performance improves automatically with experience. Finally, probabilistic modelling up to this point has been based on Gaussian distributions. Chapter 12 shows that for the hardest tracking problems, involving dense background clutter, non-Gaussian models are essential. They can be applied via random sampling algorithms, at increased computational cost, but to very considerable effect in terms of enhanced robustness.

As far as writing conventions go, references to books and papers have been kept out of the main text, to improve readability, and collected in separate bibliographic notes, appearing at the end of each chapter. These notes give sources for the ideas introduced in the body of the text and pointers to references on related ideas. Again for readability, mathematical derivations are kept from intruding on the main text by the use of two devices. The most important derivations are sandwiched (stealing a convention from Knuth's TEX manual) between

double-bend and

all-clear

road signs in the margins. These are optional reading for those who want the mathematical details. Still more optional are the results and proofs in appendix B which support chapter 9 on dynamical models.

Web page

A web page for the book is at URL `http://www.robots.ox.ac.uk/~contours/` and contains MPEG sequences and additional material for those interested in exploring further the ideas discussed in the book.

1.2 Applications

A decade ago, it seemed unlikely that the research effort invested in Computer Vision would be harvested practically in the foreseeable future. Partly this reflected the lack of computational power of hardware available at the time, a limitation which has been greatly eased by the passing years. Partly though it was the result of an ambitious view of the problems of vision, in which the aim was to build a general purpose vision engine, rather than particular applications. More recently, that view has been rather overtaken by a more focused, algorithmically driven approach. The result is that Computer Vision ideas are working their way into a variety of practical applications, particularly in the areas of robotics, medical imaging and video technology.

The active contour approach is a prime candidate for practical exploitation. This is because active contours make effective use of specific prior information about objects and this makes them inherently efficient algorithms. Furthermore, active contours apply image processing selectively to regions of the image, rather than processing the entire image. This enhances efficiency further, allowing, in many cases, images to be processed at the full video rate of 50/60 Hz. Incidentally, the ability to do vision at real-time rate has an important spin-off in stiffening criteria of acceptability, amounting to a qualitative re-evaluation of standards. As an example, an algorithm that locates the outline of a mouth in a single image nine times out of ten might be considered quite successful. Let loose on a real-time image sequence of a talking mouth, this is re-interpreted as abject failure — the mouth is virtually certain to be "lost" within a second or so, and the loss is usually unrecoverable. The ability to follow the mouth while it speaks an entire paragraph, tracking through perhaps 1000 video frames is an altogether more stringent test.

Ten examples of applications follow. Earlier ones are already promising candidates for commercial application while later ones are more speculative.

Actor-driven facial animation

A deforming face is reliably tracked to relay information about the variation over time of expression and head position to a Computer Graphics animated face. The relayed expression can be reproduced or systematically exaggerated. Tracking can be accomplished in real time, keeping pace with rate of acquisition of video frames so the actor can be furnished with valuable visual feedback. Systems currently available commercially rely on markers affixed to the face. Visual contour tracking allows

marker-free monitoring expression, given a modicum of make-up applied to the face, something to which actors are well accustomed. An example of real-time re-animation is illustrated for a cartoon cat in figure 1.2. This was done using two SGI INDY workstations, linked by network, one for visual tracking and one for mapping tracked motion onto the cat animation channels and display.

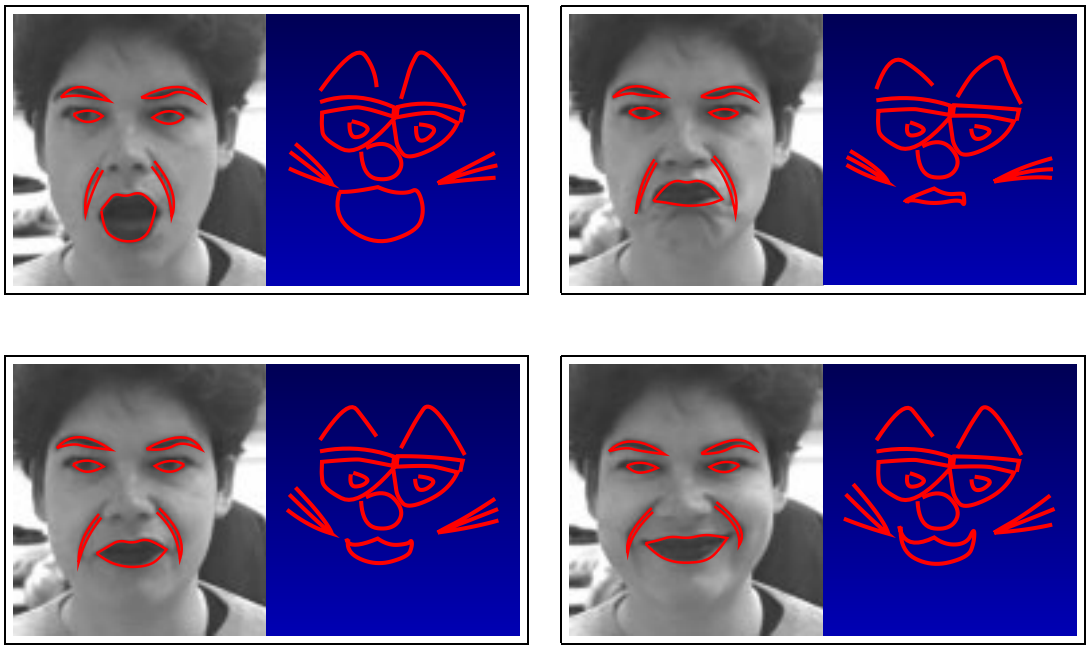


Figure 1.2: Actor-animated cat. *Tracked facial motions drive input channels to a cartoon cat, programmed with some exaggeration of expression. (Figure courtesy of Benedicte Bascle, Ben North and Julian Morris.)*

Traffic monitoring

Roadside video cameras are already familiar in systems for automated speed checks. Overhead cameras, sited on existing poles, can relay information about the state of traffic — its density and speed — and anomalies in traffic patterns. Contour tracking is particularly suited to this task because vehicle outlines form a tightly constrained class of shapes, undergoing predictable patterns of motion. Already the state of California has sponsored research leading to successful prototype systems.

Work in our laboratory, monitoring the motion of traffic along the M40 motorway near Oxford, is illustrated in figure 1.3. Vehicle velocity is estimated by recording

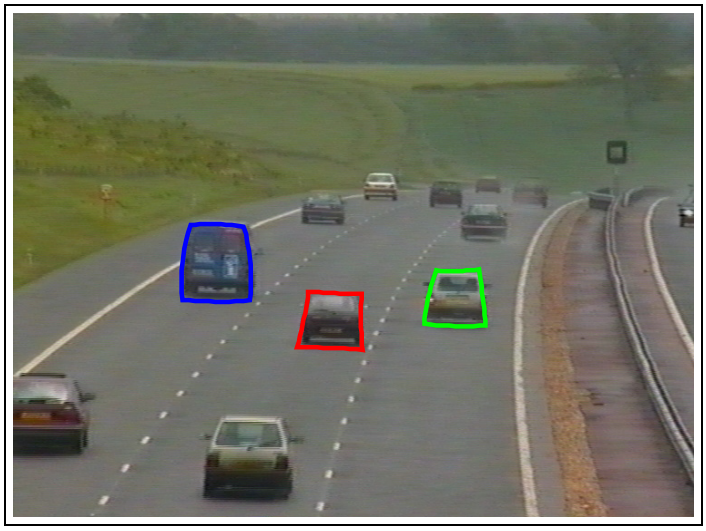


Figure 1.3: Traffic monitoring. *By automatically tracking cars, the emergency services can, for example, obtain rapid warning of an accident or traffic jam. (Illustration taken from (Ferrier et al., 1994).)*

the distance traversed by the base of a tracked vehicle contour over a known elapsed time. The measured distance is in image coordinates and this must be converted to world coordinates to give true distance. The mapping between coordinate systems is determined as a projective mapping between the image plane and the ground plane. The mapping is calibrated in standard fashion from the corners of a rectangle on the ground of known dimensions (known by reference to roadside markers which are standard fittings on British motorways), and the corresponding rectangle in the image plane, as in figure 1.4. Analysis of speeds shows clearly a typical pattern of UK motorway traffic with successively increasing vehicle speeds towards the centre lanes of the carriageway. This is summarised in the table in figure 1.5.

Automatic crop spraying

Agricultural systems for crop spraying suffer from limited ability to control overspray. Excess fertiliser seeps into the water table, a problem that is increasingly becoming a

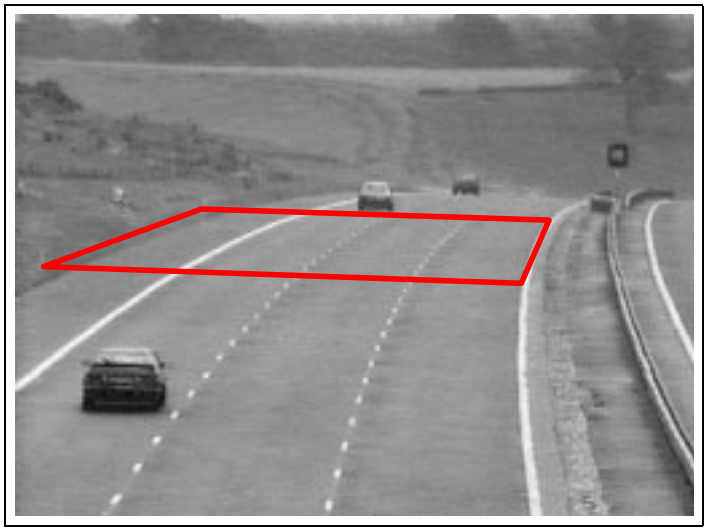


Figure 1.4: Calibration of the image-ground mapping. *Positions of the four corners of a known rectangle on the ground and its projection onto the image plane are sufficient to determine the mapping, using standard projective methods. (Illustration taken from (Ferrier et al., 1994).)*

target of legislators. It is clearly also highly desirable to ensure that toxic chemicals used to control weeds are directed away from plants intended for human consumption. Segmentation of video images on the basis of colour can be an effective means of visually separating plant from soil but is disrupted by shadows cast by the moving tractor. Contour tracking, as in figure 1.6, offers an alternative means of detecting plants that is somewhat immune to such disruption.

Robot grasping

The use of vision in robotics is commonplace in commercial practice, both for inspection and for coordination of grasp. Figure 1.7 shows an experimental system designed for use with a camera mounted on the robot's wrist, to determine stable two-fingered grasps. A snake is used to capture the outline shape, and geometric calculations along the B-spline curve, using first and second derivatives to calculate orientation and curvature, establish a set of safe grasps.

region (lane)	start (sec)	exit (sec)	distance (yards)	speed (mph)	av spd (mph)
1	269.28	273.96	132	58	
1	275.92	279.72	127	68	
1	297.86	301.56	129	72	
1	303.96	308.40	130	60	68
1	314.12	317.24	133	87	
1	321.76	325.24	126	74	
1	330.20	334.04	132	70	
1	343.16	347.58	123	57	
2	687.38	692.18	158	67	
2	708.46	712.36	164	86	
2	727.26	731.20	155	80	76
2	733.12	737.72	164	73	
2	749.12	753.64	169	77	
3	506.78	510.66	156	83	79
3	513.04	517.04	148	75	

Figure 1.5: Analysis of data from tracked cars. *Vehicle velocities are measured between gates space 150 yards apart. (Data from experiments reported in (Ferrier et al., 1994).)*

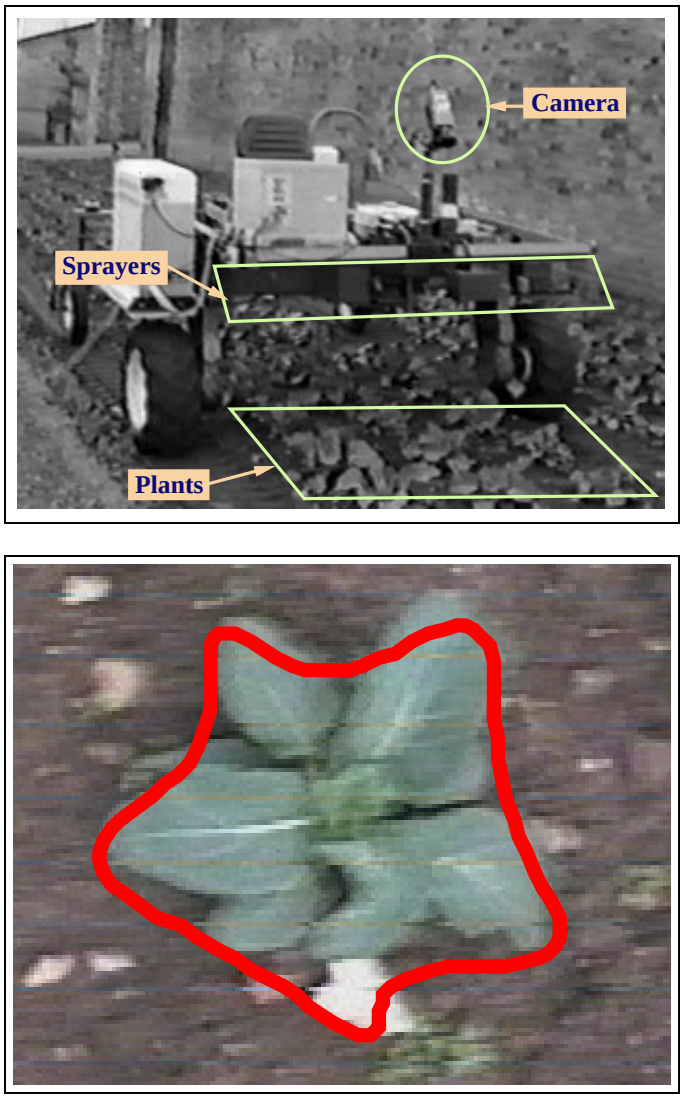


Figure 1.6: Robot tractor. *An autonomous tractor carrying a camera and computer for video analysis has the task of spraying earth and plants automatically, using an array of independently controlled spray nozzles. Plants can be segmented dynamically from the earth and weeds around it, the spraying of fertiliser and weed-killer to be directed onto or away from plants as appropriate. (Figures courtesy of David Reynard, Andrew Wildenberg and John Marchant.)*

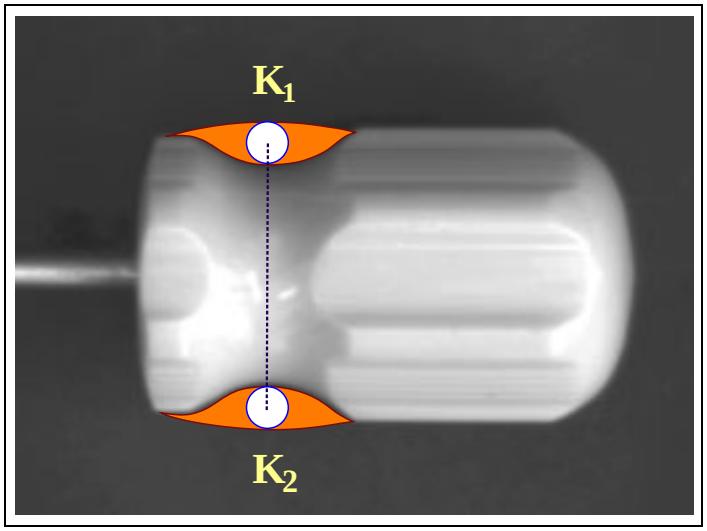


Figure 1.7: Robot hand-eye coordination. *The white circles are placement regions for two thin fingers, computed automatically from the outline of the screwdriver handle. Provided each finger lies within its circle, closing the gripper is bound to capture the screwdriver. (Figure courtesy of Colin Davidson.)*

Surveillance

A combination of visual motion sensing and contour tracking is used to follow an intruder on a security camera in figure 1.8. The camera is mounted on a computer controlled pan-tilt platform driven by visual feedback from the tracked contour.

Biometrics: body motion

This application (figure 1.9) involves the measurement of limb motion for the purposes of analysis of gait as a tool for planning corrective surgery. The tool is also useful for ergonomic studies and anatomical analysis in sport. It is related to the facial animation application above, but more taxing technically. Again, marker based systems exist and are commercially successful as measurement tools both in biology and medicine but it is attractive to replace them with marker-free techniques. There are also increasingly applications in Computer Graphics for whole body animation. Capture of the motion of an entire body from its outline looks feasible but several problems remain to be solved: the relatively large number of degrees of freedom of the articulating body

Figure 1.8: Tracking a potential intruder on security video. *(Figure courtesy of Simon Rowe, David Murray.)*

poses stability problems for trackers; the agility of, say, a dancing figure requires careful treatment of "occlusion" — periods during which some limbs and body parts are obscured by others.

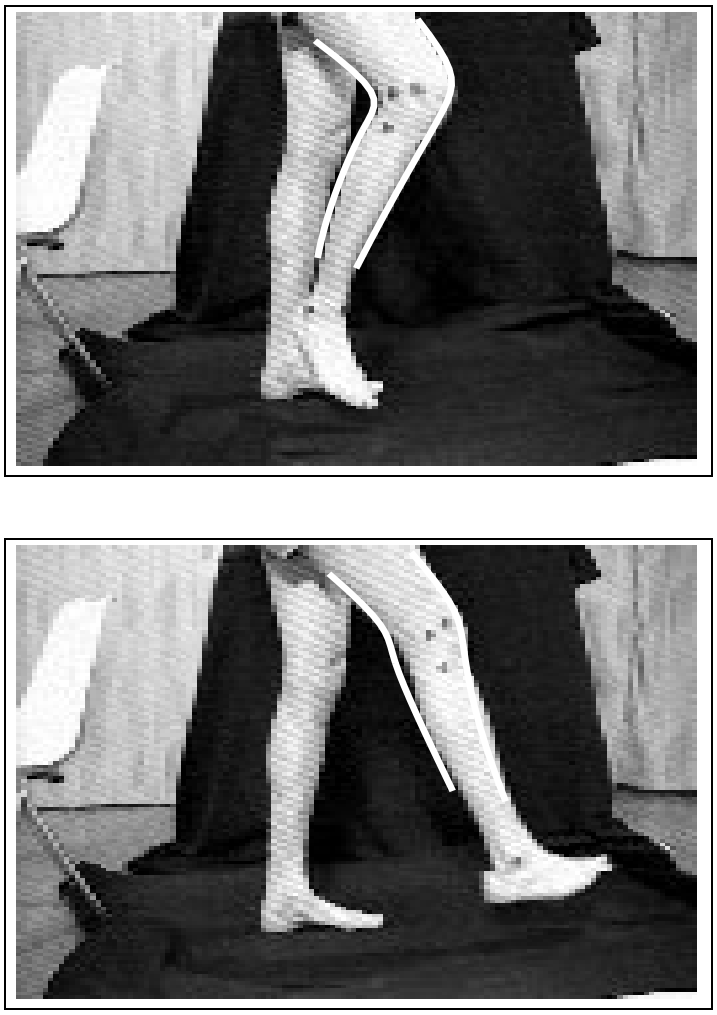


Figure 1.9: Biometrics. *Tracking the articulated motion of a human body is applicable both to biometrics and clinical gait analysis and for actor-driven whole body animation. (Figure courtesy of Rupert Curwen and Julian Morris.)*

Audio-visual speech analysis

Automatic speech-driven dictation systems are now available commercially with large vocabularies though often restricted to separately articulated words. The functioning of such a system is dependent on very reliable recognition of a small set of keywords. In practice, adequately reliable keyword recognition has been realised in low-noise environments but is problematic in the presence of background noise, especially cross-talk from other speakers. Independent experiments in several laboratories have suggested that lip-reading has an important role to play in augmenting the acoustic signal with independent information that is immune to cross-talk. Active contour tracking has been shown to be capable of providing this information (figures 1.10 and 1.11), robustly and in real time, resulting in substantial improvements in recognition-error rates.

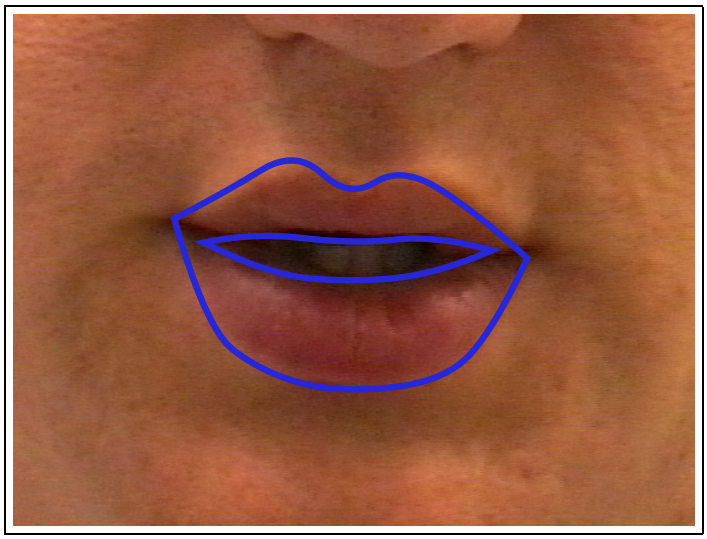


Figure 1.10: Speech-reading. *Performance in automatic speech recognition can be enhanced by lip-reading. This is done by tracking visually the moving outlines of lips to obtain visual signals which are synchronised with the acoustic signal. (Figure courtesy of Robert Kaucic and Barney Dalton.)*

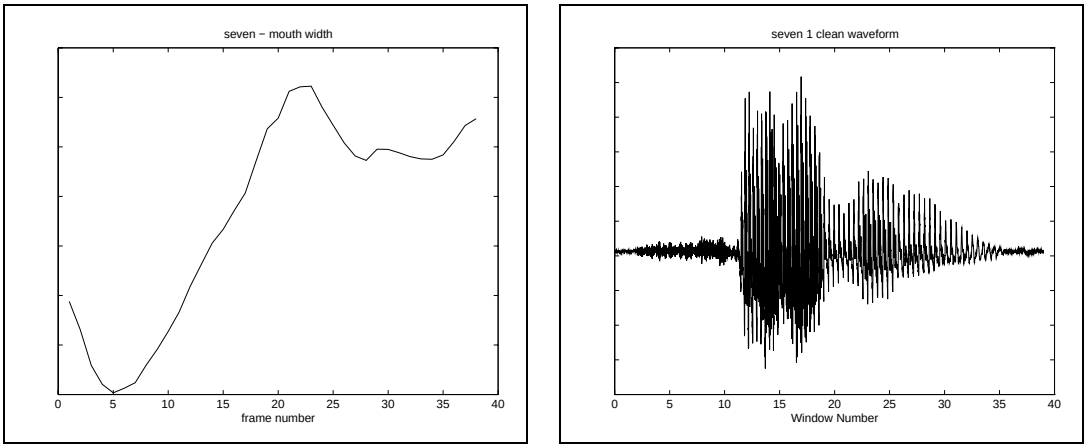


Figure 1.11: Audio and visual speech signals *This figure shows visual (left) and audio (right) signals for the spoken word "seven," over a duration of 0.6 s.*

Medical diagnosis

Ultrasound scanners are medical diagnostic imaging devices that are very widely available owing to their low cost. They are especially suited to dynamic analysis owing to their ability to deliver real-time video sequences. There are numerous potential applications for automated analysis of the real-time image sequences, for example the analysis of abnormalities in cardiac action as in figure 1.12. Noisy artifacts — ultrasound speckle — make these images especially hard to analyse. In this context, active contours are particularly powerful because speckle-induced error tends to be smoothed by the averaging along the contour that is a characteristic of active contour fitting. Broadly tuned, learned models of motion are used in tracking as prior constraints on the moving subject, to aid automated perception. The research issue here is how to learn more finely tuned models to classify normal and aberrant motions.

Another important imaging modality for medical applications is "Magnetic Resonance Imaging" (MRI). It is an expensive technology, but popular because it is as benign as ultrasound, yet as detailed as tomographic X-rays. Applications are pervasive, and one specific example concerning measurements of the cerebral hemispheres of the brain is illustrated in figure 1.13. In each of successive slices of the brain image, two separate snakes lock onto the outlines of the left and the right hemispheres. Geometric coherence in successive slices means that a fitted snake from one slice can

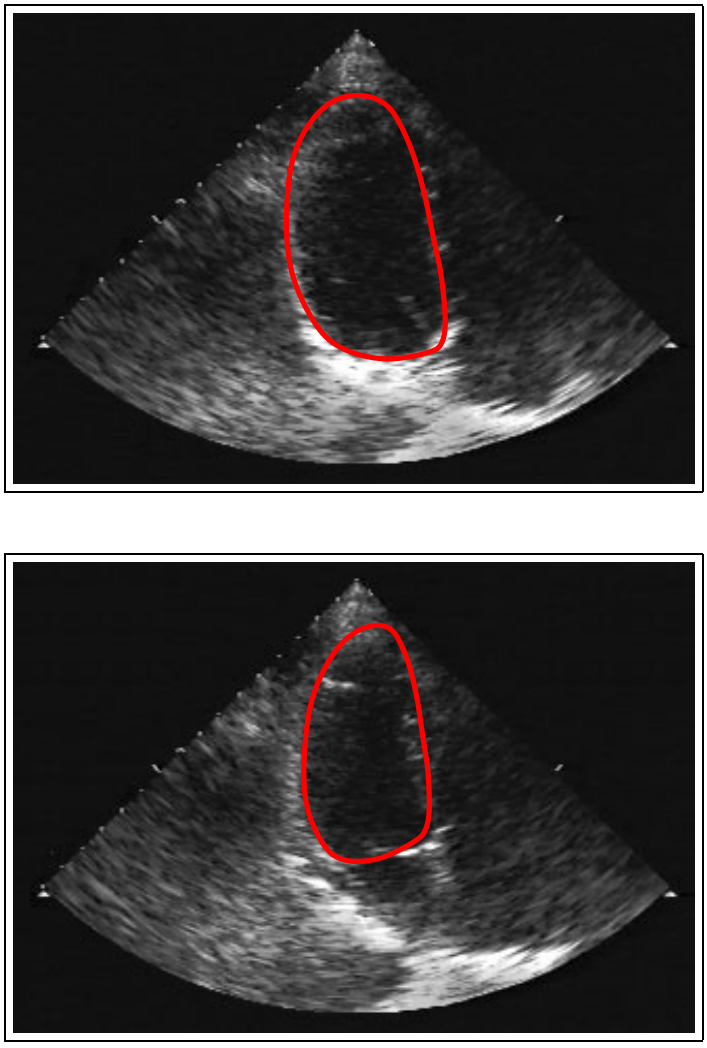


Figure 1.12: Medical echocardiogram analysis. *The left ventricle beating heart is tracked by ultrasound imaging for use in medical diagnosis. (Figure courtesy of Alison Noble and Gary Jacob.)*

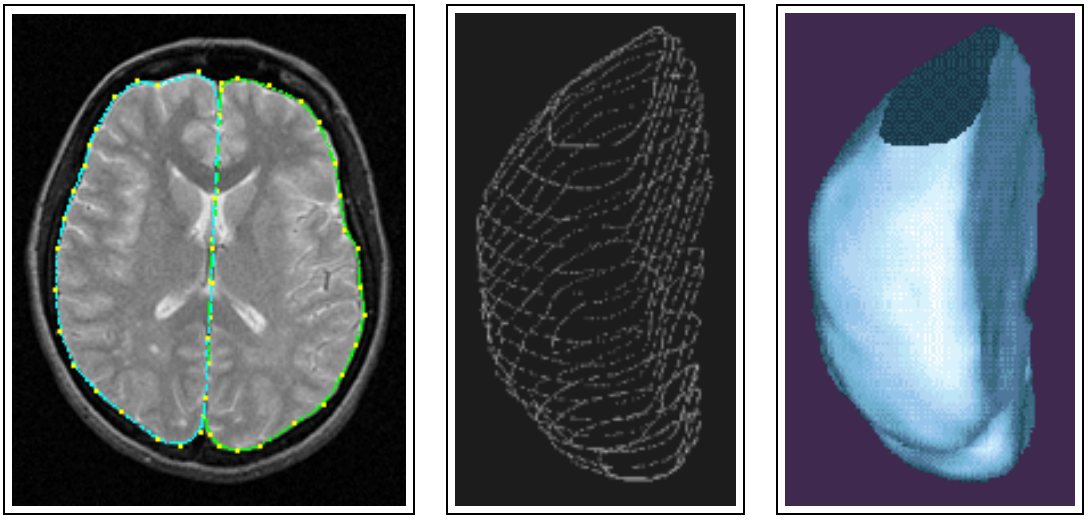


Figure 1.13: MRI imaging of brain hemispheres. *Each MRI scan (left) of the brain images one cross-sectional slice of the brain. Separate snakes trace outlines of the left and right hemispheres. Slices from one hemisphere are stacked (middle), converted to a mesh and finally rendered as a solid (right). (Figures reproduced from (Marais et al., 1996).)*

be used as the initial snake for the next. The entire fitting process can therefore be initialised by hand fitting snakes around outlines in the first slice. The degree of symmetry of the reconstructed hemispheres has been proposed as a possible diagnostic indicator for schizophrenia.

Automated video editing

It is standard practice to generate photo-composites by "blue-screening" in which a foreground object, photographed in motion against a blue background is isolated electronically. It can then be superimposed against a new background to create special effects. Contour tracking raises the possibility of doing this with objects photographed against backgrounds that have not been prepared specially in any way, as in figure 1.14. This increases the versatility of the technique and raises the possibility of extracting moving objects from existing footage for re-incorporation in new video sequences. In a second example (figure 1.15), the motion of a cluster of leaves is not only tracked, but also interpreted as a three-dimensional displacement, so that a computer-generated

Figure 1.14: Automated video editing. *Tracking the outline of a foreground object allows it to be separated automatically from the background, and manipulated as desired, a special effect which can otherwise only be achieved by "blue-screening" from specially prepared footage.*

object can be "hung" from the cluster and added to the animation. This is achieved despite the heavy clutter in the background that makes tracking harder by tending to camouflage the moving leaves.

User interface

The use of body parts as input devices for graphics has of course been thoroughly explored in "Virtual Reality" applications. Current devices such as data-gloves and infra-red helmets are cumbersome and restrictive to the wearer. Visual tracking technology raises the possibility of flexible, non-contact input devices as in figure 1.16. One aim is to use tracking to realise the "digital desk" concept in which a user manipulates a mixture of real and virtual documents on a desk, the virtual ones generated by an overhead video-projector.

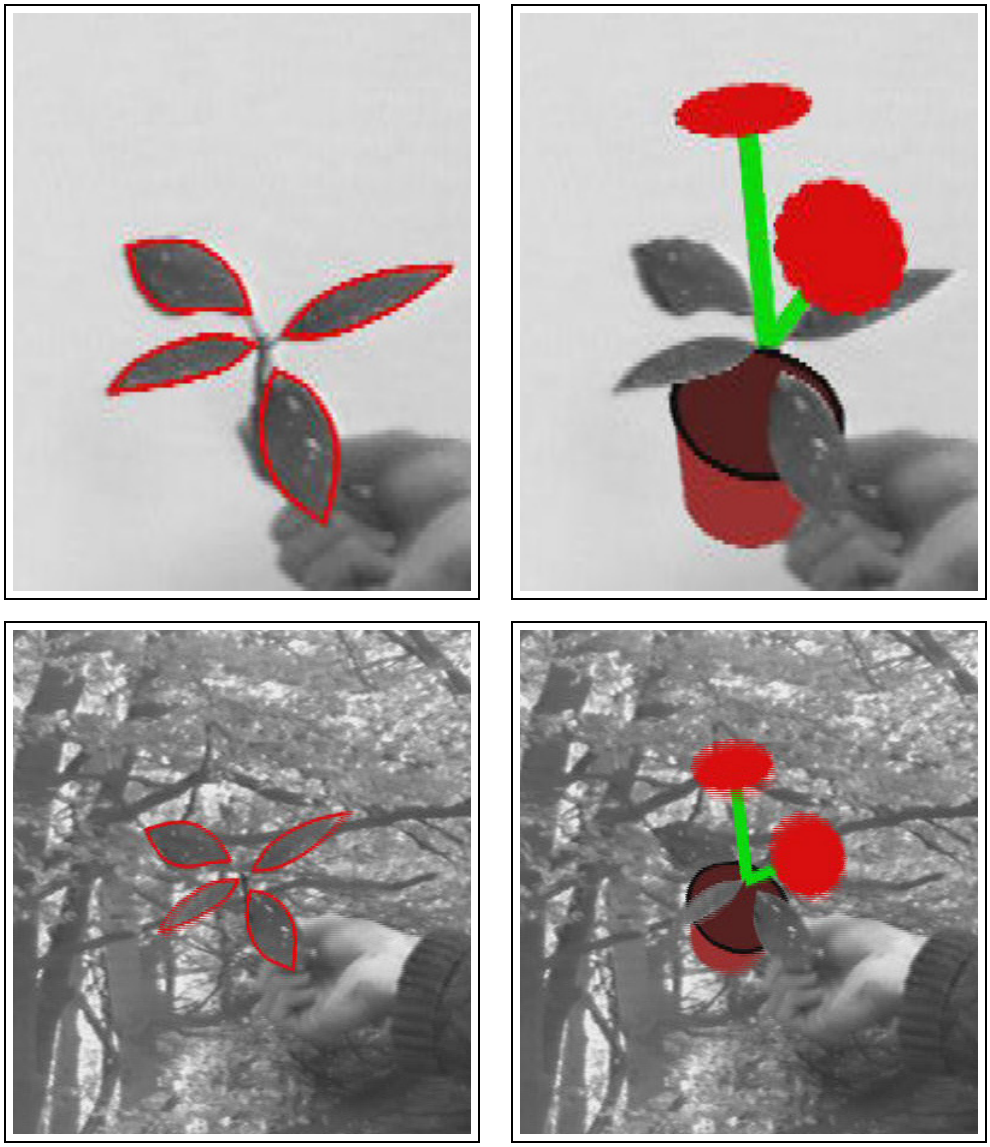
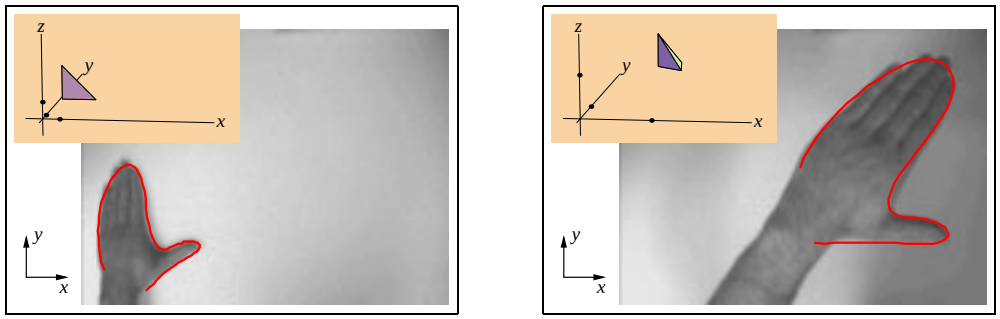
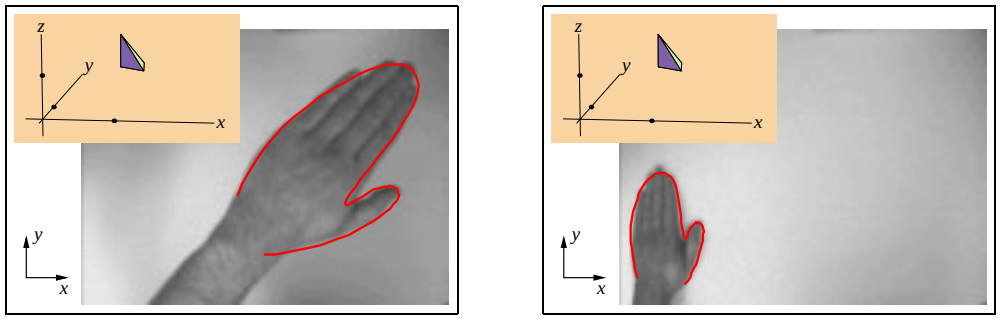


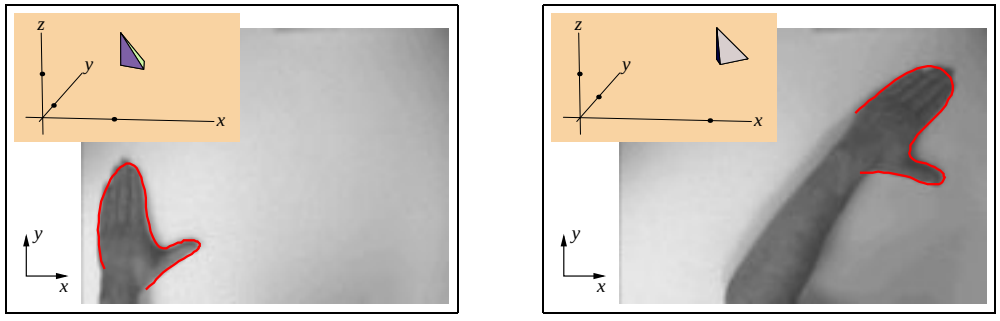
Figure 1.15: Automated re-animation. *A cluster of leaves is tracked as it moves (top), its motion interpreted three-dimensionally, and computer-generated pot and flowers are added. This technique is then applied to a sequence with the leaf cluster moving against heavy clutter (bottom).*



Hand translates in x, y and z directions and rotates; object follows hand's motion.



Thumb closed to "lock" object while hand returns to start.



Thumb open: object follows hand translating and rotating.

Figure 1.16: *A hand tracked in real time by a video camera acts as a three-dimensional mouse. Moving the thumb towards the hand acts as an "indexing" gesture, equivalent to lifting a conventional mouse off the desk to reposition it without moving the pointer. (Figure courtesy of Ben North.)*

Bibliographic notes

Despite enormous research effort, the pinnacle of which is represented by (Marr, 1982), the goal of defining general low-level processes for vision has proved obstinate and elusive. Much effort was directed towards finding significant features in images. The theory and practice of image-feature detection is very fully developed — some of the landmarks include (Roberts, 1965; O'Gorman, 1978; Haralick, 1980; Marr and Hildreth, 1980; Canny, 1986; Perona and Malik, 1990) on feature detection and (Montanari, 1971; Ramer, 1975; Zucker et al., 1977) on grouping them into linear structures. See also (Ballard and Brown, 1982) for a broad review. The challenge lies in recovering features undamaged and free of breaks, and in successfully grouping them according to the object to which they belong. In some cases subsequent processes can tolerate errors — gaps in contours and spurious fragments — and this is particularly true of certain approaches to object recognition, for instance (Ballard, 1981; Grimson and Lozano-Perez, 1984; Faugeras and Hebert, 1986; Mundy and Heller, 1990). Another important theme in "low-level" vision has been matching using features, including (Baker and Binford, 1981; Buxton and Buxton, 1984; Grimson, 1985; Ohta and Kanade, 1985; Pollard et al., 1985; Ayache and Faverjon, 1987; Belhumeur, 1993), mostly applied to matching pairs of stereoscopic images.

One notably successful reaction against the tyranny of low-level vision was "active vision" (Aloimonos et al., 1987; Bajcsy, 1988) whose progress and achievements are reviewed in (Blake and Yuille, 1992; Aloimonos, 1993; Brown and Terzopoulos, 1994). Another radical departure was the "snake", for which the original paper is (Kass et al., 1987), and many related papers are given in the bibliography to the following chapter. Pattern theory is a general statistical framework that is important in the study of active contours. It was developed over a number of years by Grenander (Grenander, 1981), and a lucid summary and interpretation can be found in (Mumford, 1996). Again, many related papers following the pattern theory approach are given in the course of the book.

Applications

Actor-driven animation is a classic application for virtual reality systems. Tracking of changing expressions can be done using VR hardware, or visually with reflective markers (Williams, 1990), using active contours (Terzopoulos and Waters, 1990; Terzopoulos and Waters, 1993; Lanitis et al., 1995) or using so-called "optical flow" (Essa

and Pentland, 1995; Black and Yacoob, 1995). Underlying muscular motion may be modelled to constrain tracked expressions.

Traffic monitoring is firmly established as a viable application for machine vision, for traffic information systems, non-contact sensors, and autonomous vehicle control (Dreschler and Nagel, 1981; Dickmanns and Graefe, 1988a; Sullivan, 1992; Dickmanns, 1992; Koller et al., 1994; Ferrier et al., 1994). Projective (homogeneous) transformations (Mundy and Zisserman, 1992; Foley et al., 1990) are used for the conversion between image and world coordinates.

Automated crop-handling based on vision has become a realistic possibility in the last decade (Marchant, 1991; Plá et al., 1993), and active contour tracking has a role to play here (Reynard et al., 1996).

A series of theories of determining stable grasps based on an outline have been proposed (Faverjon and Ponce, 1991; Blake, 1992; Rimon and Burdick, 1995a; Rimon and Burdick, 1995b; Rimon and Blake, 1996; Ponce et al., 1995; Davidson and Blake, 1998) and are particularly suited to real-time grasp planning with active contours (Taylor et al., 1994).

A pioneering advance in the visual tracking of human motion was Hogg's "Walker" (Hogg, 1983) which used an articulated model of limb motion to constrain search for body parts. Active contours have been applied with some success to tracking whole bodies and body parts (Waite and Welsh, 1990; Baumberg and Hogg, 1994; Lanitis et al., 1995; Goncalves et al., 1995), though methods based on point features can also be useful for coarse tracking (Rao et al., 1993; Murray et al., 1993).

Audio-visual speech analysis, or speech-reading, has been the subject of psychological study for some time (Dodd and Campbell, 1987). The computational problem has received a good deal of attention recently, using both active contours (Bregler and Konig, 1994; Bregler and Omohundro, 1995; Kaucic et al., 1996) and methods based more directly on image intensities (Petajan et al., 1988), or using artificial facial markers (Finn and Montgomery, 1988; Stork et al., 1992). Generally, as in conventional speech recognition, Hidden Markov Models (HMMs) (Rabiner and Bing-Hwang, 1993) are used for classification of utterances, e.g. (Adjoudani and Benoit, 1995).

Several researchers have investigated the application of active contours to the interpretation of medical images, for example (Amini et al., 1991; Ayache et al., 1992; Cootes et al., 1994).

The technique of rotoscoping allows film-makers to transfer a complex object from one image sequence to another. This can be done automatically using blue-screening (Smith, 1996) if the object can be filmed against a specially prepared background.

Computer-aided techniques for object segmentation are also of great interest for augmented reality systems, which attach computer-generated imagery to real scenes. Traditionally mechanical or magnetic 3D tracking devices have been used (Grimson et al., 1994; Pelizzari et al., 1993; Wloka and Anderson, 1995) to solve this problem, but they are inaccurate and cumbersome. Vision-based tracking has been used instead (Kutulakos and Valliano, 1996; Uenohara and Kanade, 1995; State et al., 1996; Heuring and Murray, 1996), especially for medical applications, mostly restricted to tracking artificial markers. Graphical objects can be made to pass behind real ones (State et al., 1996), by building models of the real-world objects off-line, using scanned range maps.

Effective ways of using a gesturing hand as an interface are yet to be generally established. One very appealing paradigm is the "digital desk" (Wellner, 1993) in which moving hands interact both with real pieces of paper and with virtual (projected) ones, on the surface of a real desk. Other body parts may also be useful for controlling graphics, for instance head (Azarbayejani et al., 1993) and eyes (Gee and Cipolla, 1994). Gestures need not only to be tracked but also interpreted by classifying segments of trajectories, either in configuration space or phase space (Mardia et al., 1993; Campbell and Bobick, 1995; Bobick and Wilson, 1995). This is related both to classification of speech signals (see above) and to classification of signals in other domains, such as electro-encephalograph (EEG) in sleep (Pardey et al., 1995).

Chapter 2

Active shape models

Active shape models encompass a variety of forms, principally *snakes*, *deformable templates* and *dynamic contours*. Snakes are a mechanism for bringing a certain degree of prior knowledge to bear on low-level image interpretation. Rather than expecting desirable properties such as continuity and smoothness to emerge from image data, those properties are imposed from the start. Specifically, an elastic model of a continuous, flexible curve is imposed upon and matched to an image. By varying elastic parameters, the strength of prior assumptions can be controlled. Prior modelling can be made more specific by constructing assemblies of flexible curves in which a set of parameters controls kinematic variables, for instance the sizes of various subparts and the angles of hinges which join them. Such a model is known as a deformable template, and is a powerful mechanism for locating structures in an image.

Things become more difficult when it is necessary to locate moving objects in image sequences — the problem of tracking. This calls for dynamic modelling, for instance invoking inertia, restoring forces and damping, another key component of the original snake conception. We refer to curve trackers that use prior dynamical models as "dynamic contours." Later parts of the book are all about understanding, specifying and learning dynamical prior models of varying strength, and applying them in dynamic contour tracking.

2.1 Snakes

The art of feature detection has been much studied (see bibliographic notes for previous chapter). The principle is that a "mask" or "operator" is designed which produces an output signal which is greatest wherever there is a strong presence in an image of a feature of a particular chosen type. The result is a new image or "feature map" which codes the strength of response for the chosen feature type, at each pixel. Examples of feature maps for three different kinds of feature are illustrated in figure 2.1. Details of the designs of masks and the application to images by digital convolution are given in chapter 5. For now it is sufficient to say that the operator is a sub-image which is scanned over an image using "mathematical correlation" or "convolution" (this is explained in chapter 5). The mask is a prototype image, typically of small size, of the feature being sought: for a valley feature, for instance, the mask would be a V-shaped intensity function. The output of the correlation process is a measure of goodness of fit of the prototype to the image, in each of the image locations evaluated.

However, feature maps are only the beginning. They enhance features of the desired type but do not unambiguously detect them. Detection requires a decision to be made at each pixel, the simplest decision rule being that a feature is marked wherever feature strength exceeds some preset threshold. A constant threshold is rarely adequate except for the simplest of situations such as an opaque object on a back-lit table, as commonly used in machine vision systems. However, the features on a face cannot be back-lit and, if the threshold is set high, gaps appear in edges. If the threshold is low, spurious edges appear, generated by fine texture. Often no happy medium exists. More subtle decision schemes than simple thresholds have been explored but after around two decades of concerted research effort, one cannot expect to do very much better than the example in figure 2.2. The main structure is present but the topology of hand contours is disrupted by gaps and spurious fragments.

The lesson is that "low-level" feature detection processes are effective up to a point but cannot be expected to retrieve entire geometric structures. Snakes constitute a fundamentally new approach to deal with these limitations of low-level processing. The essential idea is to take a feature map $F(\mathbf{r})$ like the ones in figure 2.1, and to treat $(-F(\mathbf{r}))$ as a "landscape" on which the snake, a deformable curve $\mathbf{r}(s),\ 0 \leq s \leq 1$, can slither. For instance, a filter that gives a particularly high output where image contrast is high will tend to attract a snake towards object edges. Equilibrium equations for $\mathbf{r}(s)$ are set up in such a way that $\mathbf{r}(s)$ tends to cling to high responses of F, that is, maximising $F(\mathbf{r}(s))$ over $0 \leq s \leq 1$, in some appropriate sense. This ten-

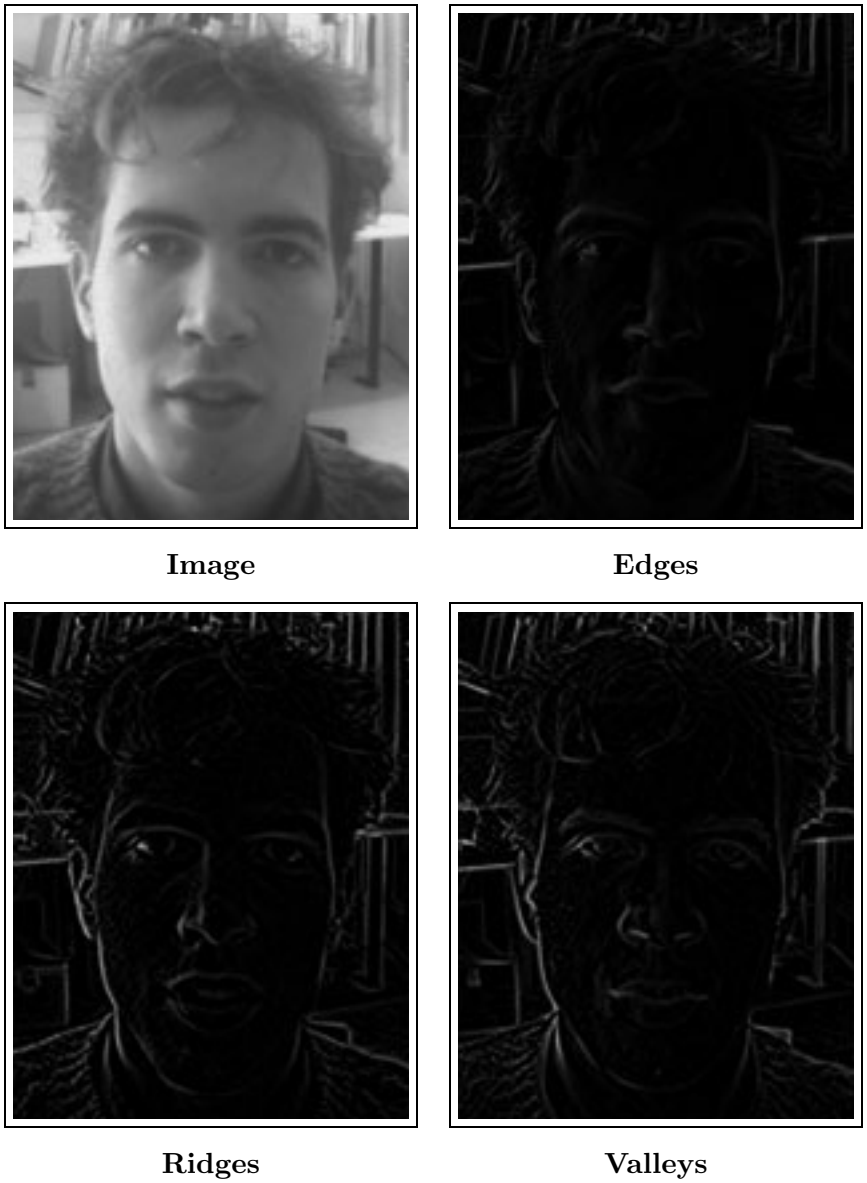


Figure 2.1: Image-feature detectors. *Suitably designed image filters can highlight areas of an image in which particular features occur. The examples shown here filter for areas of high contrast ("edges"), peaks of intensity ("ridges") and intensity troughs ("valleys").*

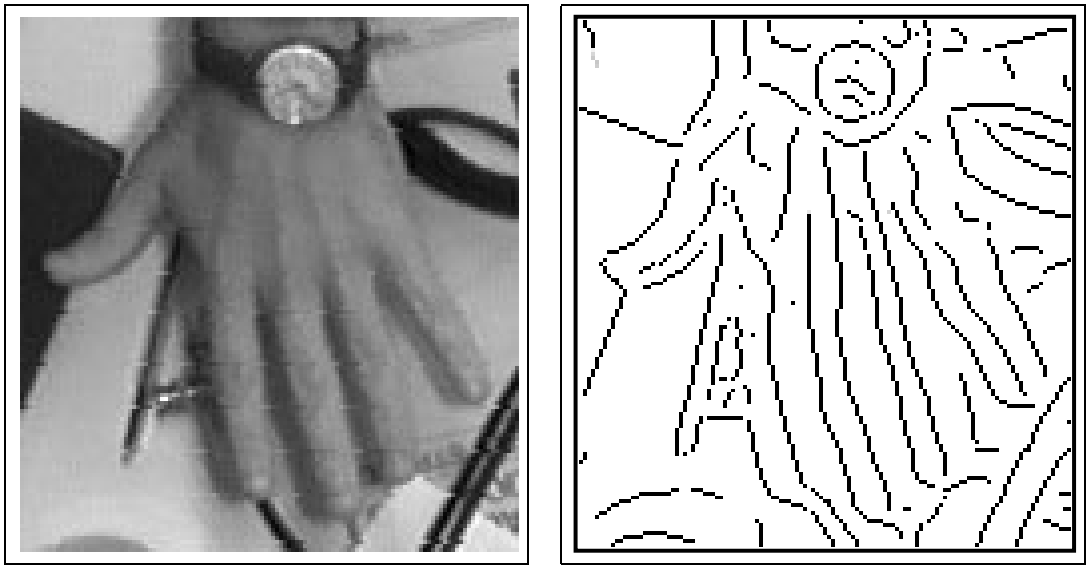
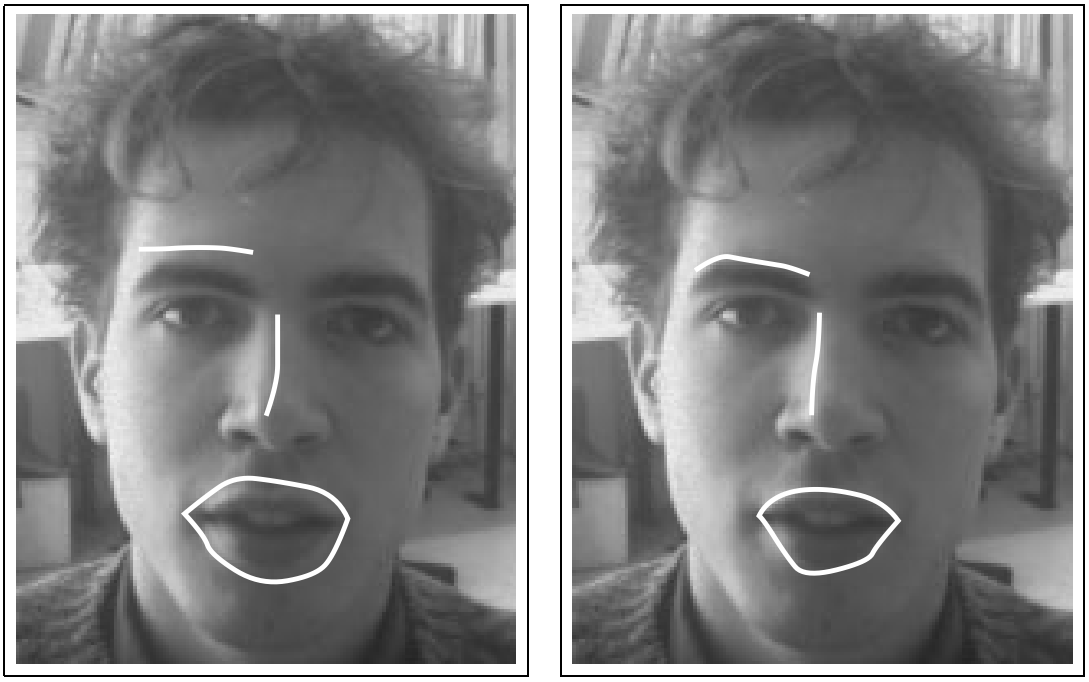


Figure 2.2: Detecting edges. *Edges (right) are generated from the image (left) using horizontally and vertically oriented masks and a decision process (Canny, 1986) that attempts to repair gaps. Nonetheless, there are breaks at critical locations such as corners or junctions, and spurious fragments that disrupt the topology of the hand.*

dency to maximise F is formalised as the "external" potential energy of the dynamical system. It is counterbalanced by "internal" potential energy which tends to preserve smoothness of the curve. The equilibrium equation is:

$$\underbrace{\left(\frac{\partial(w_1\mathbf{r})}{\partial s} - \frac{\partial^2(w_2\mathbf{r})}{\partial s^2}\right)}_{\text{internal forces}} + \underbrace{\nabla F}_{\text{external force}} = 0. \tag{2.1}$$

(Note: $_s$ and $_t$ subscripts denote differentiation with respect to space and time, and ∇F is the spatial gradient of F.) If (2.1) is solved iteratively, from a suitable configuration, it will tend to settle on a ridge of the feature map F, and figure 2.3 illustrates this. The coefficients w_1 and w_2 in (2.1), which must be positive, govern the restoring forces associated with the elasticity and stiffness of the snake respectively. Either of these coefficients may be allowed to vary with s, along the snake. For example, allowing w_2 to dip to 0 at a certain point $s = s_0$ will allow the snake to kink there, as illustrated at the mouth corners in figure 2.3. Increasing w_2 encourages the snake to be smooth,



Initial configuration **Final configuration**

Figure 2.3: Snake equilibrium. *Snakes are shown in initial and final configurations. The eyebrow snake moves over an edge-feature map. The mouth snake is also attracted to edge-features; smoothness constraints are suspended at mouth corners, to allow the snake to kink there. Given that the strongest feature on the nose is a ridge (see figure 2.1), the nose snake is chosen to be attracted to ridges.*

like a stiff but flexible rod, but also increases its tendency to regress towards a straight line. Increasing w_1 makes the snake behave like stretched elastic which encourages an even parameterisation of the curve, but increases the tendency to shortness, even collapsing to a point unless counterbalanced by external energy or constraints.

Discrete approximation

Practical computations of $\mathbf{r}(s)$ must occur over discrete time and space, and approximate the continuous trajectories of (2.1) as closely as possible. The original snake represented $\mathbf{r}(s)$ by a sequence of samples at $s = s_i,\ i = 1, \ldots, N$, spaced at intervals

of length h, and used "finite differences" to approximate the spatial derivatives $\mathbf{r}_s$ and $\mathbf{r}_{ss}$ by

$$\mathbf{r}_s(s_i) = \frac{\mathbf{r}(s_i) - \mathbf{r}(s_{i-1})}{h} \quad \text{and} \quad \mathbf{r}_{ss}(s_i) = \frac{\mathbf{r}(s_{i+1}) - 2\mathbf{r}(s_i) + \mathbf{r}(s_{i-1})}{h^2}$$

and solve the resulting simultaneous equations in the variables $\mathbf{r}(s_1), \ldots, \mathbf{r}(s_N)$. The system of equations is "sparse," so that it can be solved efficiently, in time $O(N)$ in fact.

In finite difference approximations, the variables $\mathbf{r}(s_i)$ are samples of the curve $\mathbf{r}(s)$, at certain discrete points, conveying no information about curve shape between samples. Modern numerical analysis favours the "finite element" method in which the variables $\mathbf{r}(s_i)$ are regarded as "nodal" variables or parameters from which the continuous curve $\mathbf{r}(s)$ can be completely reconstructed. The simplest form of finite-element representation for $\mathbf{r}(s)$ is as a polygon with the nodal variables as vertices. Smoother approximations can be obtained by modelling $\mathbf{r}(s)$ as a polynomial "spline curve" which passes near but not necessarily through the nodal points. This is particularly efficient because the spline maintains a degree of smoothness, a role which otherwise falls entirely on the spatial derivative terms in (2.1). The practical upshot is that with B-splines the smoothness terms can be omitted, allowing a substantial reduction in the number of nodal variables required, and improving computational efficiency considerably. For this reason, the B-spline representation of curves is used throughout this book. Details are given in chapter 3.

Robustness and stability

Regularising terms in the dynamical equations are helpful to stabilise snakes but are rather restricted in their action. They represent very general constraints on shape, encouraging the snake to be short and smooth. Very often this is simply not enough, and more prior knowledge needs to be compiled into the snake model to achieve stable behaviour. Consider the following example in which a snake is set up with internal constraints reined back to allow the snake to follow the complex outline of the leaf in figure 2.4. In fact it is realised as a B-spline snake with sufficient control points to do justice to the geometric detail of the complex shape. Suppose now the snake is required to follow an image sequence of the leaf in motion, seeking energy minima repeatedly, on successive images in the sequence. If all those control points are allowed to vary somewhat freely over time, the tracked curve can rapidly tie itself into unrecoverable

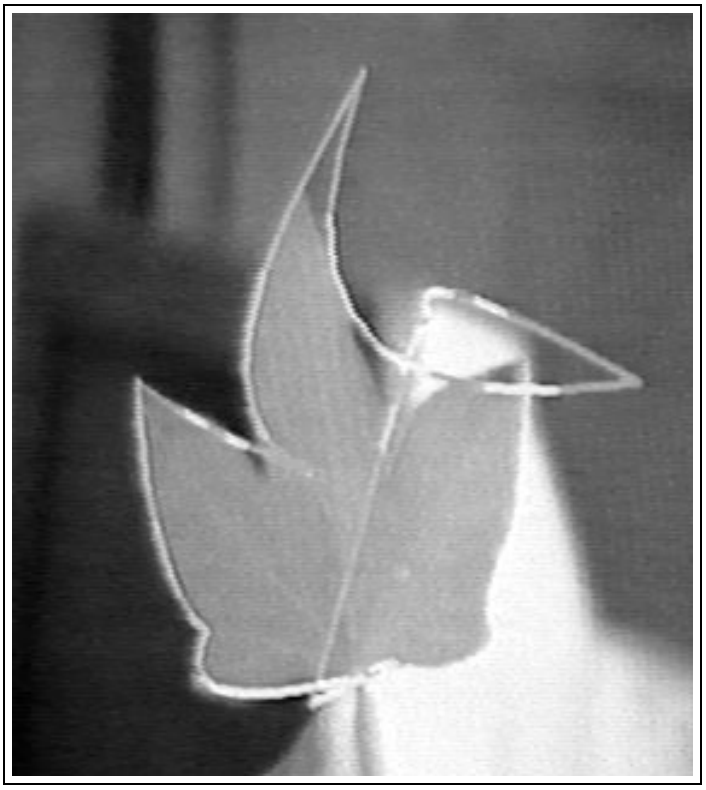


Figure 2.4: The need for shape-spaces. *The white curve is a B-spline with sufficient control points to do justice to the complexity of the leaf's shape. Control point positions vary over time in order to track the leaf outline. However, if the curve momentarily loses lock on the outline it rapidly becomes too tangled to be able to recover. (Figure by courtesy of R. Curwen.)*

knots, as the figure shows. This is a prime example of the sort of insight that can be gained from real-time experimentation. A regular snake, with suitably chosen internal energy may succeed in tracking several dozen frames off-line. However, once tracking is seen as a *continuous* process, and this is the viewpoint that real-time experiments enforce, the required standards of robustness are altogether more stringent. What was an occasional failure in one computation out of every few, becomes virtually certain eventual failure once the real-time process is allowed to run. It is of paramount importance that recovery from transients — such as a gust of wind causing the leaf to twitch — is robust.

This need for robustness is what drives the account of active contours given in this book. General mechanisms for setting internal shape models are not sufficient. Finely tunable mechanisms are needed, representing specific prior knowledge about classes of objects and their motions. The book aims to give a thorough understanding of the components of such models, initially in geometric terms, and later in terms of probability, as a means of describing *families* of plausible shapes and motions.

2.2 Deformable templates

The prior shape constraints implicit in a snake model are soft, encouraging rather than enforcing a particular class of favoured shapes. What is more, those favoured shapes have rather limited variety. For example, in the case that $w_1 = 0$ in (2.1)), they are solutions of

$$\mathbf{r}_{ss} = 0$$

which are simply straight lines. Models of more specific classes of shapes demand some use of hard constraints, and "default" shapes more interesting than a simple straight line. This can be achieved by using a parametric shape-model $\mathbf{r}(s; \mathbf{X})$, with relatively few degrees of freedom, known as a "deformable template." The template is matched to an image, in a manner similar to the snake, by searching for the value of the parameter vector $\mathbf{X}$ that minimises an external energy $E_{\text{ext}}(\mathbf{X})$. Internal energy $E_{\text{int}}(\mathbf{X})$ may be included as a "regulariser" to favour certain shapes.

As an example of a deformable template, Yuille and Hallinan's eye template is illustrated in figure 2.5, showing how the template is parameterised, and results of fitting to an image of a face. The template $\mathbf{r}(s; \mathbf{X})$ has a total of 11 geometric parameters in the parameter vector $\mathbf{X}$ and it varies non-linearly with $\mathbf{X}$. The non-linearity is evident because, for example, one of the parameters is an angle θ whose sine and cosine appear in the functional form of $\mathbf{r}(s; \mathbf{X})$. The bounding curves of the eye are parabolas which also vary non-linearly, as a function of length parameters a, b and c. The internal energy $E_{\text{int}}(\mathbf{X})$ is a quadratic function of $\mathbf{X}$ that encourages the template to relax back to a default shape. The external energy $E_{\text{ext}}(\mathbf{X})$ comprises a sum of various integrals over the image-feature maps for edges, ridges and valleys. Each integral is taken over one of the two regions delineated by the eye model or along a template curve, which causes E_{ext} to vary with $\mathbf{X}$. Finally the total energy is minimised by iterative, non-linear gradient descent which will tend to find a good minimum, in the sense of giving a good fit to image data, provided the initial configuration is not too

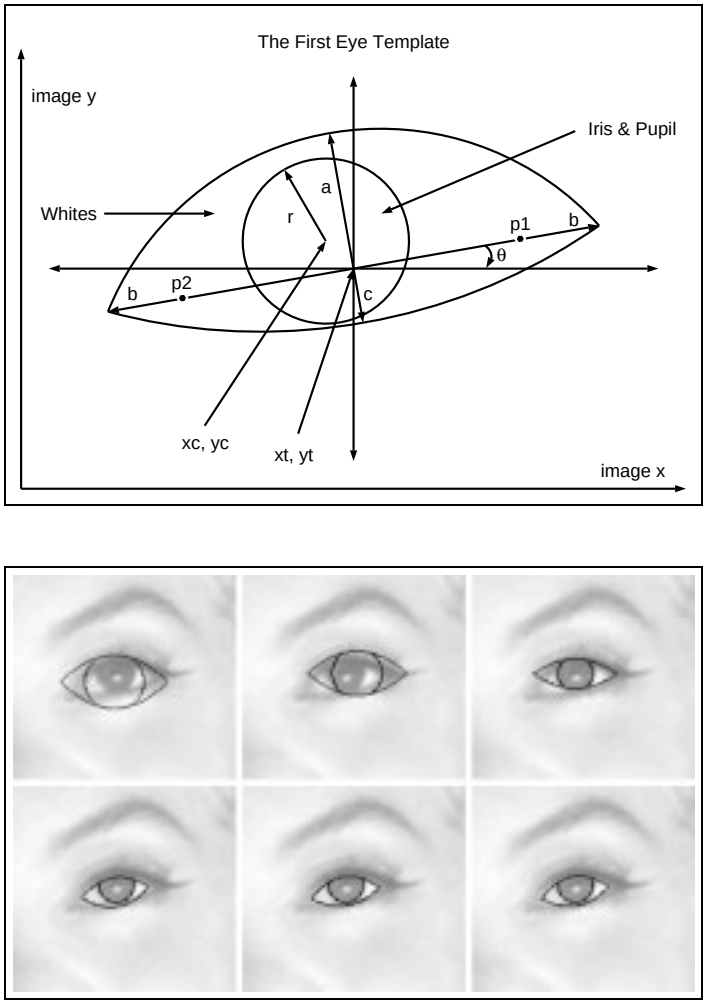


Figure 2.5: Deformable eye template *An eye template is defined (top) in terms of a modest number of variable geometric parameters. In successive iterations of a "gradient descent" algorithm, an equilibrium configuration is reached in which the template fits the eye closely. (Figure reprinted from (Yuille and Hallinan, 1992) which also gives details of external and internal energy functions.)*

far from the desired final fit.

A methodology for setting up linearly parameterised deformable templates — we term them "shape-spaces" — will be described in chapter 4. Restriction to linear parameterisation has certain advantages in simplifying fitting algorithms and avoiding problems with local minima. It is nonetheless surprisingly versatile geometrically. It should be pointed out that some elegant work has been done with three-dimensional parametric models (see bibliographic notes) but this is somewhat outside the scope of this book. Here we deal with three-dimensional motion by modelling directly its effects on image-based contour models using "affine spaces" amongst other devices.

2.3 Dynamic contours

Active contours can be applied either statically, to single images, or dynamically, to temporal image sequences. In dynamic applications, an additional layer of modelling is required to convey any prior knowledge about likely object motions and deformations. Now both the active contour $\mathbf{r}(s,t)$ and the feature map $F(t)$ vary over time. The contour $\mathbf{r}(s,t)$ is drawn towards high responses of $F(t)$ as if it were riding the crest of a wave on the feature map. The equation of motion for such a system extends the snake in (2.1) with additional terms governing inertia and viscosity

$$\underbrace{\rho\,\mathbf{r}_{tt}}_{\text{inertial force}} = -\underbrace{\left(\gamma\mathbf{r}_t - \frac{\partial(w_1\mathbf{r})}{\partial s} + \frac{\partial^2(w_2\mathbf{r})}{\partial s^2}\right)}_{\text{internal forces}} + \underbrace{\nabla F}_{\text{external force}}. \tag{2.2}$$

This is Newton's law of motion for a snake with mass, driven by internal and external forces. New coefficients in (2.2), in addition to w_1 and w_2 the elastic coefficients from (2.1), are ρ the mass density and γ the viscous resistance from a medium surrounding the snake. Given that all coefficients are allowed to vary spatially, there is clearly considerable scope for setting them to impose different forms of prior knowledge. The spatial variation also introduces a multiplicity of degrees of freedom and potentially complex effects. One of the principal aims of the book is to attain a detailed understanding of those effects, and to harness them in the design of active contours.

Most powerful of all is to combine dynamical modelling as in (2.2) with the rich geometrical structures used in deformable templates, and this is the basis of the dynamic contour. It involves defining parameterised shapes $\mathbf{r}(s;\mathbf{X})$ as for deformable

templates and then specifying a dynamical equation for the shape parameter **X**. In the dynamic contour equation (2.2), prior constraints on shape and motion were implicit, but to facilitate systematic design it is far more attractive that they should be explicit. This can be achieved by separating out a dynamical model for likely motions from the influence of image measurements. The dynamic contour becomes a two-phase process in which a dynamical model is used for *prediction*, to extrapolate motion from one discrete time to the next. Then the predicted position for each new time-step is refined using measured image features, as in figure 2.6. The "Kalman filter" is a

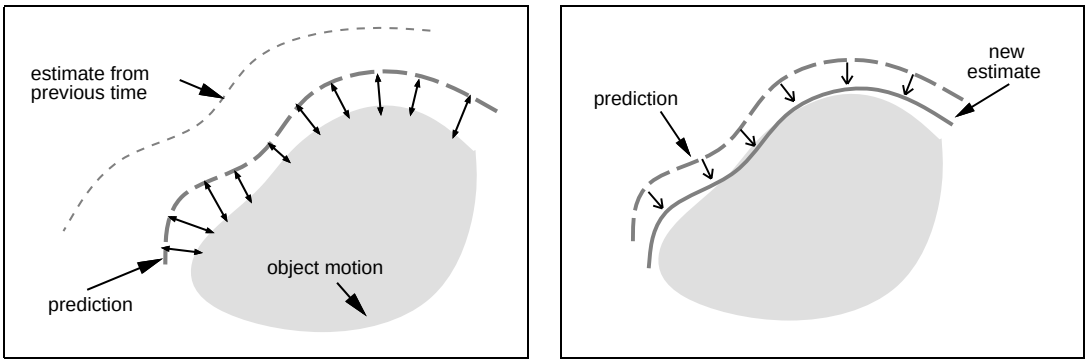


Figure 2.6: Prediction and measurement. *Dynamic contour tracking involves a two-phase process at each successive time. Past motion history and prior knowledge of motion are extrapolated to predict the displacement between successive times, then predicted position is refined using image features.*

ready made engine for applying the two-phase cycle, and for this reason has been a very popular and successful paradigm for tracking (see bibliographic notes). It is a probabilistic mechanism and this is one reason that probabilistic modelling pervades the treatment of the second part of this book.

Intuitively, predictive models demand probabilistic treatment in order to avoid being too strong. The two-phase cycle fuses a prediction with some measurements. If the prediction were deterministic with no allowance for uncertainty, it would dominate the measurements, which would therefore be ignored. As an example, consider the task of tracking a pendulum in motion. If the pendulum is believed to be executing perfect harmonic motion, free of external disturbances, then provided initial conditions are known, the future motion of the pendulum is entirely determined. Knowing initial conditions, any subsequent observation of the pendulum is redundant. Realistic

visual tracking problems are more like observing a pendulum oscillating in a turbulent airflow. The mean behaviour of the pendulum may be explained as deterministic simple harmonic motion, but the airflow drives the motion with random external forces. In terms of the shape parameter $\mathbf{X}$, this implies a dynamical equation of the form

$$\ddot{\mathbf{X}} = f(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{w}), \tag{2.3}$$

where $\dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$ are the first and second temporal derivatives of $\mathbf{X}$ and $\mathbf{w}$ is a random disturbance. Thus the value of initial conditions weaken over time, as the motion of the pendulum is progressively perturbed away from the ideal deterministic motion. This increasing uncertainty generates a "gap" in information which sensory observations can fill. A primary aim of the book is to define design principles for probabilistic models of shape and motion and explain those principles in terms of their effects both on representation of prior knowledge and in constraining and conditioning tracking performance.

Bibliographic notes

The seminal paper on snakes is (Kass et al., 1987). This spawned many variations and extensions including the use of Fourier parameterisation (Scott, 1987), incorporation of hard constraints (Amini et al., 1988) and incorporation of explicit dynamics (Terzopoulos and Waters, 1990; Terzopoulos and Szeliski, 1992). Realisation of snakes using B-splines was developed by (Cipolla and Blake, 1990; Menet et al., 1990; Hinton et al., 1992) and combined with Lagrangian dynamics in (Curwen et al., 1991). B-splines used in this way are a form of "finite element," a standard technique of numerical analysis for solving differential equations by computer (Strang and Fix, 1973; Zinkiewicz and Morgan, 1983).

The idea of deformable templates predates the development of snakes (Fischler and Elschlager, 1973; Burr, 1981; Bookstein, 1989) but has enjoyed a revival inspired by the snake. Variations on the deformable template theme rapidly emerged (Yuille et al., 1989; Yuille, 1990; Bennett and Craw, 1991; Yuille and Hallinan, 1992; Hinton et al., 1992; Cootes and Taylor, 1992; Cootes et al., 1993; Cootes et al., 1995). A good deal of research has been done on matching with three-dimensional models, both rigid (Thompson and Mundy, 1987; Lowe, 1991; Sullivan, 1992; Lowe, 1992; Harris, 1992b; Gennery, 1992) and deformable (Terzopoulos et al., 1988; Terzopoulos and Fleischer, 1988; Cohen, 1991; Terzopoulos and Metaxas, 1991; Rehg and Kanade, 1994) but

is somewhat outside the scope of this book. As models become more detailed, and search becomes more exhaustive, the three-dimensional approach merges into visual object recognition (Grimson, 1990).

The Kalman filter (Gelb, 1974; Bar-Shalom and Fortmann, 1988) is very widely used in control theory and for target tracking (Rao et al., 1993) and sensor fusion (Hallam, 1983; Durrant-Whyte, 1988; Hager, 1990) and has become a standard tool of computer vision (Ayache and Faugeras, 1987; Dickmanns and Graefe, 1988b; Dickmanns and Graefe, 1988a; Matthies et al., 1989; Deriche and Faugeras, 1990; Harris, 1992b; Terzopoulos and Szeliski, 1992; Faugeras, 1993).

Finally, it seems appropriate at least to give some pointers to approaches to visual tracking that are rather outside the active contour paradigm.

- (Black and Yacoob, 1995) uses the visual motion field over a region to track and identify movement
- (Bray, 1990) tracks using a mixture of polyhedral, model-based vision to initialise and optic-flow vectors along contours for incremental displacement
- (Fischler and Bolles, 1981; Gee and Cipolla, 1996) are very elegant uses of random generation and testing of point-correspondence hypotheses, respectively for static and dynamic image matching problems
- (Huttenlocher et al., 1993) used the "Hausdorff metric" to match successive views in a sequence; the beauty of the approach is that it requires almost no prior model of shape or motion
- (Allen et al., 1991; Papanikolopoulos et al., 1991; Mayhew et al., 1992; Brown et al., 1992; Murray et al., 1992; Heuring and Murray, 1996) are control theoretic approaches to visual-servoing, real-time tracking with robot hands and heads