Start by doing what's necessary, then do what's possible, and suddenly you are doing the impossible.

Saint Francis of Assisi

1 Introduction

Chapter Overview

T an introduction to computational shape analysis. Starting with some considerations about computer vision and 2D shapes, it proceeds by illustrating some typical applications and discussing the main problems normally involved in shape analysis, and concludes by presenting the organization of the related topics in the book chapters.

1.1 Introduction to Shape Analysis

There is little doubt that one of the most stimulating research fields, from both the scientific and technological perspectives, are those related to the main human sense: *vision*. The acquisition and analysis of the visual information produced by the interaction between light and the world objects have represented powerful means through which humans and animals can quickly and efficiently learn about their surrounding environments. The advantages of this ability for survival can be immediately recognized, accounting for all efforts nature has taken in developing such flexible visual systems. As far as humans are concerned, more than 50% of their brains are somehow involved in visual information analysis, a task that underlies the majority of human daily activities. In fact, it is hard to identify which of these activities do *not* involve, either directly or indirectly, vision. Therefore, whenever necessary to automate (e.g., in case of dangerous or tedious situations) or to improve (e.g., to increase precision and repetition) human activities, effective computer vision systems become essential.

The origin of computer vision is intimately intertwined with computer history, having been motivated by a wide spectrum of important applications such as in robotics, biology, medicine, industry, security and physics, to name but a few. Such a great deal of applications is not surprising if we consider the aforementioned importance of the human vision sense. Nevertheless, though "seeing" seems to us to be *simple*, *natural* and *straightforward*, in practice the design of versatile and robust computational vision systems has proven to be difficult, and most of the flexible computer vision systems created thus far have met with limited success. In fact, vision requires real-time processing of a very large and heterogeneous data set (including shape, spatial orientation, color, texture, motion, etc.) as well as interactions with other equally important cognitive abilities, such as memory, feelings and language. Additional difficulties with the analysis of visual information derive from noise, occlusion and distortions, as well as from the fact that image formation involves mapping from a three-dimensional space (the scene) onto a two-dimensional support (the retina or the image plane in a camera, for instance), thus implying information to be lost. Notwithstanding these difficulties, there is no doubt that robust vision systems are viable, for nature has created highly adapted and efficient vision systems in so many animals. In this context, vision science has developed as an interdisciplinary research field, frequently involving concepts and tools from computer science, image processing, mathematics, physics, artificial intelligence, machine learning, pattern recognition, computer graphics, biology, medicine, neuroscience, neurophysiology, psychology and cognitive sciences. Although not always recognized, such areas have already provided computer vision with important insights. For instance, several important imaging concepts and techniques can be closely related to biologic principles, including the edge detection approach described in [Marr, 1982], the two-dimensional (2D) Gabor filter models developed in [Daugman, 1980], the artificial neural networks introduced by McCullogh and Pitts [Anderson, 1995], and the importance of high curvature points in shape perception described in [Attneave, 1954], to name but a few.

To probe further: ***Shape Theories in Biology and Psychology***

A related and interesting study topic are the theories of human shape perception. The reader is referred to [Biederman, 1985; Edelman, 1999; Hubel & Wiesel, 2005; Leyton, 1988, 1992; Perret & Oram, 1993; Poggio & Edelman, 1990; Rosin, 1993; Siddiqi & Kimia, 1995; Zeki, 2000] for further reading on this issue.

Among all different aspects underlying visual information, the *shape* of the objects certainly plays a special role, a fact that can be experienced while reading the characters on this page, which are essentially characterized by their shapes. In a sense, shapes can be thought as being the *words* of the visual language. Indeed, the prominent role of vision and shape (or its synonymous *form*) to humans has implied several visually-related terms to be incorporated into the common vocabulary, including the words trans*form*ation, in*sight* and *imag*ination, and expressions such as *lick into shape*, *take shape*, *shape up*, and *in any shape or form*. As far

Figure 1.1: *Image containing a 3D object (a cat) and respective representation in terms of its 2D silhouette.*

as the pictorial information is concerned, the particular issue of 2D shapes, i.e., shapes defined on the plane, is of paramount importance. As mentioned above, image formation often involves mapping objects from the three-dimensional (3D) space onto 2D structures, such as a retina or a CCD. It is worth noting that even the 2D-object silhouette often conveys enough information allowing the recognition of the original object, as illustrated in Figure 1.1.

This fact indicates that the 2D shape analysis methods described in this book can often be applied for the analysis of 3D objects. While there are many approaches for obtaining the full 3D representation of objects in computer vision, be it by reconstruction (from stereo, from motion, from shading, etc.) or by using special devices (e.g., 3D scanners), dealing with 3D models still is computationally expensive, frequently to a prohibitive degree, hence the importance of 2D approaches for treating such situations. Of course, there are several objects, such as characters, which are defined in terms of 2D shapes and should therefore be represented, characterized and processed as such.

In a more general situation, 2D shapes are often the archetypes of objects belonging to the same pattern class, which is illustrated in Figure 1.2.

In spite of the lack of additional important pictorial information, such as color, texture, depth and motion, the objects represented by each of the silhouettes in this image can be promptly recognized. Some of these 2D shapes are abstractions of complex 3D objects, which are represented by simple connected sets of black points on the plane (see Chapter 4 for additional discussion on the issue of shapes).

This book is precisely about obtaining, processing and analyzing shape images in automated, or at least semi-automated, fashion by using digital computers. In a

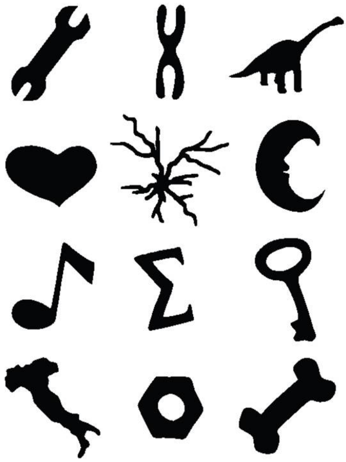


Figure 1.2: *Some typical and easily recognizable 2D shapes.*

typical application, the image of a shape is digitized, yielding a digital shape that can be pre-processed, analyzed and (eventually) classified. As mentioned above, these techniques have been successfully applied to a wide range of practical problems, some of which are exemplified in the following table. In order to gain a deeper insight about computational shape analysis, two representative applications illustrating typical situations in practical shape analysis are outlined and discussed in the next section. An introductory overview of the several tasks involved in shape analysis is presented in the remainder of this chapter.

Research Field	Examples of Applications
Neuroscience	Morphological taxonomy of neural cells, investigations about the interplay between form and function, comparisons between cells of different cortical areas and between cells of different species, modeling of biologically realistic cells, and simulation of neural structures.
Document analysis	WWW, OCR (optical character recognition), multimedia databases, and historical documents.
Visual arts	Video restoration, special effects, video tracking, games, computer graphics, visualizations, and image synthesis.
Internet	Content-based information retrieval, watermarking, graphic design, and usability.

Continued on next page.

Continuation.

Research Field	Examples of Applications
Medicine	Tumor recognition, quantification of change and/or deformation of anatomical structures (e.g., endocardial contour of left ventricle of heart, corpus callosum), morphometric analysis for diagnosis (e.g., multiple sclerosis and Alzheimer's disease), numerical analysis of chromosomes, identification of genetic pathologies, laparoscopy, and genetic studies of dentofacial morphology.
Biology	Morphometric-based evolution comparison, taxonomy, interplay between form and function, comparative anatomy, cytology, identification and counting of cells (e.g., white blood cells), characterization of cells and nuclear shapes, growth and shape modifications, analysis of human gait, analysis of electrophoretic gels, and microscopy.
Physics	Analysis of particle trajectories, crystal growth, polymers, characterization of star clusters in astronomy, and several types of microscopy.
Engineering	Semiconductors, quality control, danger detection, machine interpretation of line drawings, computer-aided design of mechanical parts and buildings, automation, robotics, remote sensing, image and video format standards, and spatial exploration.
Security	Fingerprint/face/iris detection, biometrics, human gait, and signature verification.
Agriculture	Harvest control, seed counting and quality control, species identification, and fruit maturation analysis.

To probe further: ***Shape Analysis***

The multidisciplinarity of image analysis, with respect to both techniques and applications, has motivated a rich and impressive set of information resources represented by conferences, books, WWW URLs and journals. Some of the more important of these are listed in the book Web page at: http://www.ime.usp.br/~cesar/shape_crc/chap1.html.

1.2 Case Studies

1.2.1 Case Study: Morphology of Plant Leaves

A special problem where shape analysis usually comes into play is the classification of biological entities based on respective morphological information, as illustrated in the following (see [Bruno et al., 2008a]). Figure 1.3 shows a series of 12 images of leaves obtained from four different species of plants.

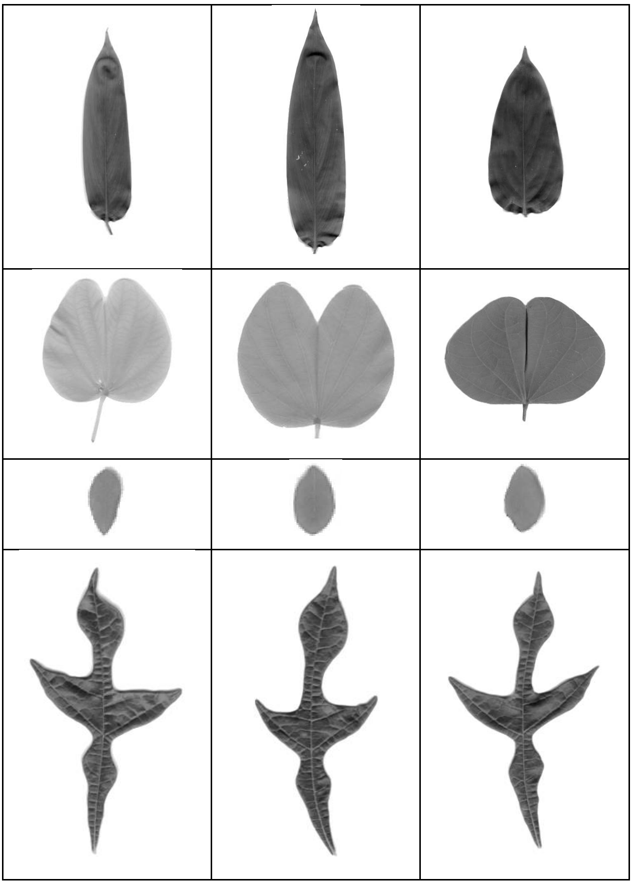


Figure 1.3: *Set of plant leaves belonging to four classes.*

Observe that in this situation the classes of leaves are clearly defined in terms of the respective plant species. Now, suppose that we want to classify an unknown leaf, i.e., to assign it to one of the four plant *classes* presented in Figure 1.3. A typical *pattern recognition* approach to solve this problem is to measure a series of *features*, or *attributes*, from each leaf image in Figure 1.3, say a feature related to the brightness distribution of each leaf and a feature related to its size or extension.

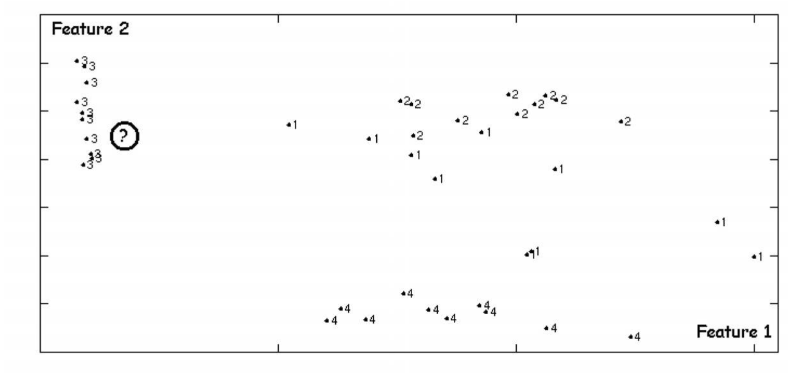


Figure 1.4: *Feature space obtained for plant leaves of the types shown in Figure 1.3.*

An example of the former feature type is the histogram entropy, denoted by f_1, which can be informally understood as a number indicating the degree of disorder of the gray levels inside each leaf (see Chapter 3 for additional discussion). The other feature could be characterized by the perimeter of each leaf, denoted by f_2. Therefore, each leaf is represented in terms of a pair (f_1, f_2), known as the *feature vector* associated with each pattern. The *feature space* of our pattern recognition problem, which is shown in Figure 1.4, is the 2D space defined by $f_1 \times f_2$ for all initially considered leaves.

In Figure 1.4, each point is labeled with its class number, i.e., 1, 2, 3 or 4, to the leaf classes of Figure 1.3. It is interesting to note that each pattern class has defined a respective *cluster* (informally speaking, a localized and separated cloud of points) in the feature space. Back to the initial problem of classifying an unknown leaf based on its image, it would be useful to have a *pattern classifier* that could produce the correct class for each supplied feature vector (f_1, f_2) corresponding to a new leaf not in the original database. For instance, in case the measured features (f_1, f_2) is that indicated by a "?" in Figure 1.4, then it would be reasonable to assume that the unknown leaf belongs to class 3, for the feature vector is much closer to that cluster than to all other remaining clusters in the feature space. This simple approach to automated classification is called, for obvious reasons, *nearest neighbor*.

To probe further: ***Applications in Agriculture***

The page `www.ee.surrey.ac.uk/Research/VSSP/demos/leaf/index.html` presents an interesting application of the leaf classification problem to agriculture.

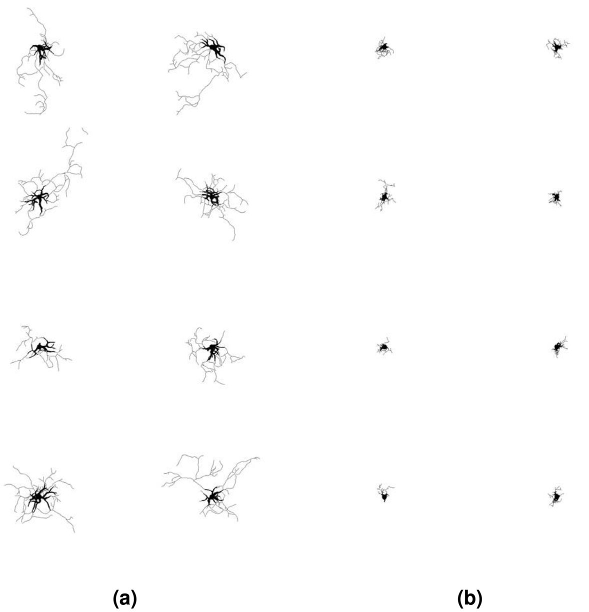


Figure 1.5: *Two morphological classes of prototypical cat ganglion cells: α-cells* (a) *and β-cells* (b). *The cells have been artificially generated by using formal grammars [Costa et al., 1999].*

1.2.2 Example: Morphometric Classification of Ganglion Cells

The second example of a practical shape analysis application concerns the morphological analysis of neural cells. The morphology of neurons has been recognized to be an extremely important property of such cells since the pioneering work of the neuroscientist [Ramon y Cajal, 1989]. The morphology has been often related to specific physiological properties of the cell (e.g., [Costa, 2005; Costa & Velte, 1999]). A specific example of this situation is defined by the two morphological classes of retinal ganglion cells in the cat, i.e., the α- and β-cells. Several studies by neuroscientists over decades have produced results clearly indicating an interesting relationship between the form and function of these cells. The morphological classes for α- and β-cells have been defined by taking into account especially the neural dendritic branching pattern, with α-cells presenting dendrites spread throughout a larger area, while the β-cells are more concentrated. Figures 1.5 (a) and (b) present two sets of α- and β-cell prototypes, respectively, clearly illustrating their shape differences.

Many are the motivations for developing objective morphological classification schemes for such neural cells. First, such objective parameters would help the creation of taxonomic classification with respect to neural morphology, as well as to review previously proposed classes typically developed in a subjective fashion (i.e., through human inspection). Furthermore, such studies can lead to advances regarding the characterization of the relationship between neural form and function, which is a particularly important problem in neuroscience. In addition, objective parameters about neural morphology are essential for paving the way towards more realistic models and computer simulations of neural structures [Ahnert & Costa, 2008; Costa et al., 1999]. Research has been carried out in order to develop quantitative measures about the geometry of neural cells (a research area called *neuromorphology*) that could properly reflect the differences between the different types of neural cells.

In this context, neuroscientists have started applying mathematical tools, such as the fractal dimension and the bending energy, in order to devise automatic tools allowing the effective classification of neurons with respect to their morphology. Nevertheless, the development of such features has met interesting and difficult problems that must be resolved by the researchers. For instance, there is no agreement among neuroscientists with respect to the number of morphological classes of neurons. In fact, the morphological classes can be defined and redefined by neuroscientists as new methods are developed. Therefore, an interesting shape analysis problem arises involving the following difficult questions:

① How many morphological classes of neurons are there?

② How can we assign cells to morphological classes?

③ What features (not only morphological, but also characterizing the respective neural activity) should we adopt in order to characterize the neural cells?

④ How reliable are the shape features and the classification?

⑤ What classification methods should we use in order to classify the neural cells with respect to the adopted features?

It is worth emphasizing that these questions are by no means restricted to the problem of morphological characterization of neural cells. In fact, many different practical situations in a wide diversity of fields face similar doubts. They are representative of both *supervised* and *unsupervised* classification schemes, as will soon become clear, and pattern recognition theory provides a set of mathematical tools that help scientists in answering (at least partially) the above questions.

1.3 Computational Shape Analysis

There are many problems usually addressed in the context of shape analysis and recognition by using computers, upon which this book is organized. In fact, computational shape analysis involves several important tasks, from image acquisition to shape classification. Figure 1.6 illustrates the shape processing tasks frequently

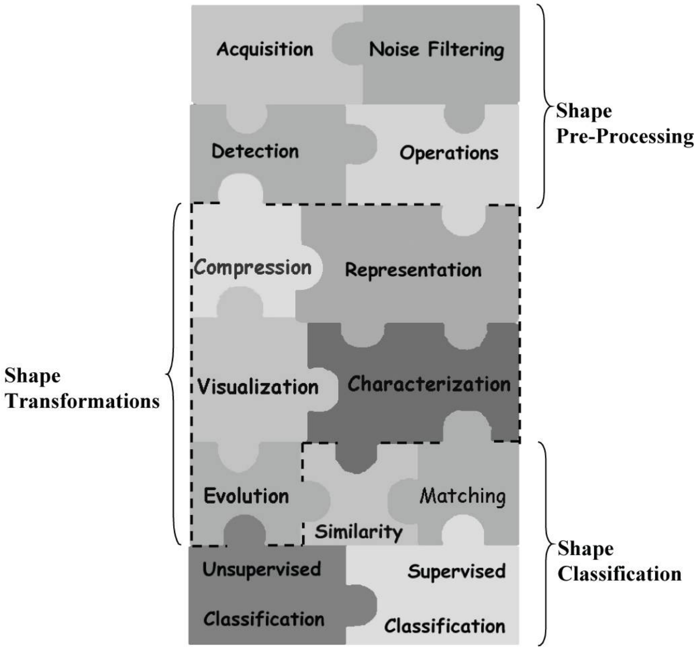


Figure 1.6: *Typical shape analysis tasks and their organization into three main classes.*

required for shape analysis, which can be broadly divided into three classes, namely *shape preprocessing*, *shape transformations* and *shape classification*.

The following sections address each of these classes of shape analysis operation:

1.3.1 Shape Pre-Processing

The first step toward the computational morphological analysis of a given object involves acquiring and storing an image of it and separating the object of interest from other non-important image structures. Furthermore, digital images are usually corrupted by noise and other undesirable effects (such as occlusion and distortions), therefore requiring the application of special procedures. The following subsections present a brief introduction to each of these problems, which are grouped together into the *shape pre-processing* category. The issues of shape acquisition and pre-processing are addressed in more detail in Chapter 3.

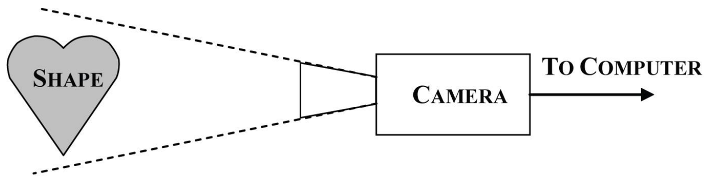


Figure 1.7: *Illustrative scheme of shape acquisition to be processed by a computer.*

Shape Acquisition

Shape acquisition involves acquiring an image (e.g., photograph) and digitizing it, so that it can be properly processed by a computer (Figure 1.7).

The type of image acquisition framework and devices to be used depend heavily on the application, e.g., a camera attached to a microscope can be used in histological applications, while a scanner can be used to acquire images of leaves, such as in the example in Section 1.2.1.

Shape Detection

One of the first steps generally required in shape analysis is to detect the shape, which involves locating the object of interest in the image so that it can be subsequently treated. For instance, Figure 1.8 shows an image including several shapes: in order to analyze them, it is necessary to locate each shape, which can have different visual properties (e.g., color and texture).

The most basic approach to shape detection is through image segmentation (e.g., by thresholding). When the image can be properly segmented so that the object of interest can be successfully isolated from other non-important image structures (including the background), then shape detection is a reasonably straightforward task. An interactive approach can also be adopted in a number of important practical situations. For instance, the object of interest can be detected by requesting a human operator to click inside the object of interest, which would be followed by a region-growing algorithm (see Chapter 3) in such a way that the resulting grown region corresponds to the detected shape. This procedure is usually implemented in most off-the-shelf image processing software, as it might have already been tried by the reader, which is usually represented as a magic wand that allows the selection of irregular image regions. However, there are many alternative approaches for object detection in digital images. For example, if the object of interest can be generally represented by a template, template-matching techniques could be applied in order to locate the object instances in the image. On the other hand, if the problem to be solved involves shape analysis in video sequences, then motion-based techniques can be used for detecting and locating the object in the image. Nevertheless, it is worth emphasizing that image segmentation can frequently become a difficult problem, mainly if the image acquisition conditions (such as illumination, camera position, focus, etc.) cannot be properly controlled.

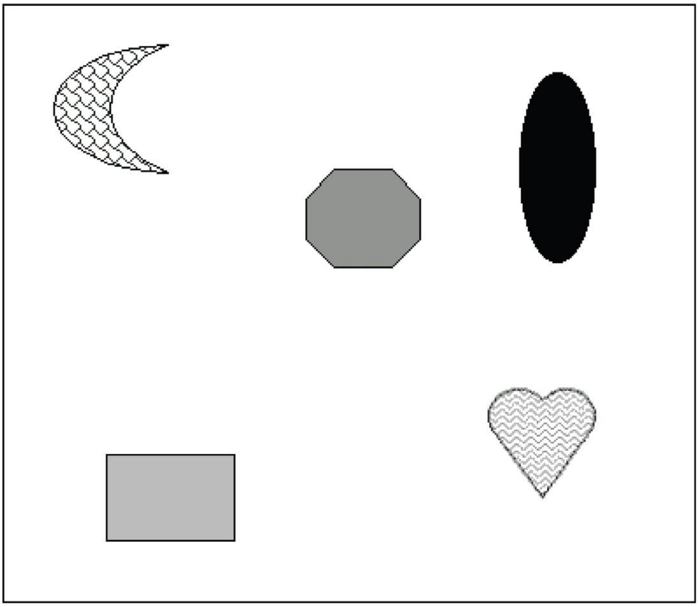


Figure 1.8: *Shape detection involves locating the objects of interest in the image. The shapes can present different visual properties, such as color and texture.*

Noise Filtering

Digital image processing systems generally have to cope with noisy images in nearly all practical situations, and shape analysis is by no means an exception: noisy shapes occur ordinarily, independently of the application (see Figure 1.9).

It is worth noting that, besides the perturbations inherent to digital images (consequences of the spatial and intensity quantizations involved in acquiring digital images), noise in shapes can also arise from the imaging operations that are typically applied in order to detect the shape of interest. Frequently the shape detection process is preceded by a series of image processing procedures, such as diverse filtering operations, data fusion and segmentation, which can introduce perturbations at nearly every processing stage. Furthermore, shape quantization and sampling, which are necessary for obtaining digital shapes, are usually a source of critical noise. All these noisy alterations are generally reflected as small modifications on the obtained shape, which can affect subsequent shape analysis procedures. Consequently, approaches to shape representation and description often attempt to be robust to noise or to incorporate some noise filtering mechanism. For instance, multiscale techniques such as curvature estimation (Chapter 7) adopt filtering as an inherent means to reduce or to eliminate quantization and other types of noise.

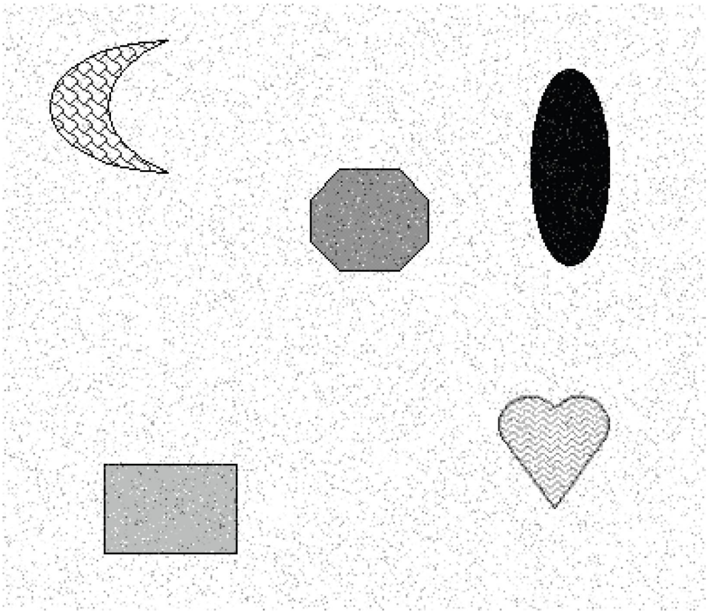


Figure 1.9: *If noise is present in the image, proper noise filtering procedures should be applied before and/or during shape analysis.*

Noise filtering is discussed in Chapter 3 and related techniques, such as multiscale curvature estimation, are discussed in Chapters 6 and 7.

Shape Operations

There are many important operations that can be applied to shapes. For instance, if the problem to be solved involves comparing two or more shapes, then they should be normalized so that the comparison makes sense (Figure 1.10).

Normalization processes usually involve parameters such as scale, rotation and translation. Shape warping, registration and morphing are also examples of shape operations that can be applied to normalization and comparison. Typically, such operations are based on defining a mapping between a set of points (*landmarks*) along two or more shapes, which allows the generation, by interpolating, of a series of intermediate shapes that would possibly be obtained while *transforming* one of the shapes into the other. Shape manipulation/handling can also include interactive edition (e.g., elimination of portions of the shape) and operations aiding visualization, as well as operations involving more than one shape (e.g., shape addition and intersection). The most important shape operations are discussed in Chapter 4.

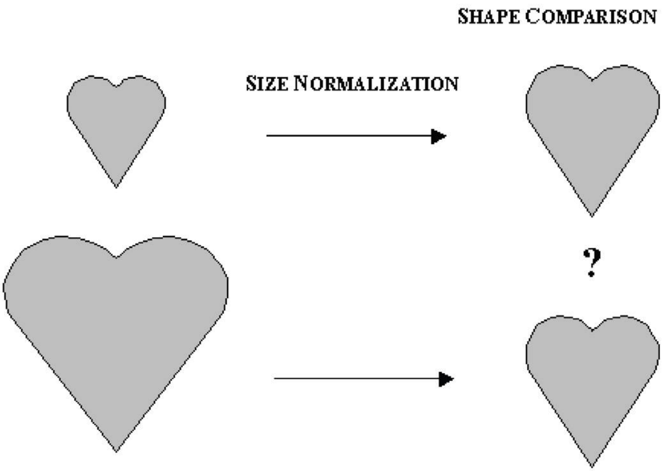


Figure 1.10: *Shape operations can be involved while normalizing some visual properties, e.g., before comparing shapes. It is important to note that, in some applications, the differences in size are actually an important shape parameter, which would make size normalization inadequate in such situations. Similar comments apply to other visual properties, such as orientation and translation normalization.*

1.3.2 Shape Transformations

Once the shape of interest has been acquired and processed (for instance, noise has been substantially reduced), a set of techniques can be applied in order to extract information from the shape, so that it can be analyzed. Such information is normally extracted by applying suitable *shape transformations*. Such transformations are mappings that allow both representation of the shape in a more appropriate manner (with respect to a specific task) and extraction of measures that are used by classification schemes. The concept of shape transformation is covered in Chapter 4, while Chapters 5, 6, and 7 present computational techniques for feature extraction.

Shape Evolution

It is often important to study the properties of a sequence of shapes corresponding to an object that has evolved along a certain time period (an example is shown in Figure 1.11).

For instance, it is important to establish the correspondences between different points of the ventricular contour as the heart beats or to analyze development of neurons and other cells as they grow. All these problems can be treated in terms of shape transformations as shape evolution.

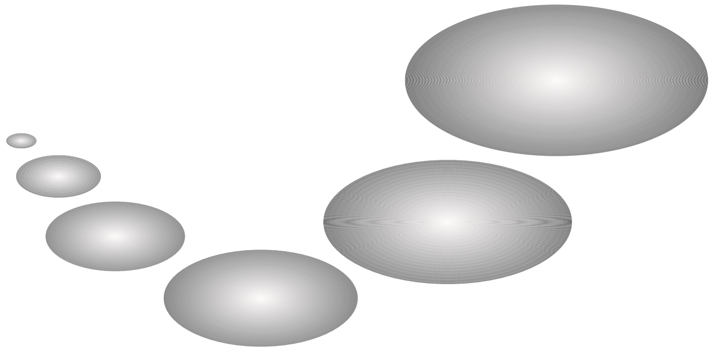


Figure 1.11: *Shape evolution involves analyzing an object that has modified its shape along a certain time period. This is often carried out by observing a series of shapes that the object has assumed during the period.*

Shape Representation

Once the object of interest has been located in the image (through shape detection and segmentation, see Section 1.3.1), its shape is understood as being formed by the set of points found to belong to that object. In this sense, the first representation of the object shape is the set of points identified in the original image. It is often the case that such representation, though naturally produced by the shape detection procedure, is not particularly useful, being even considered cumbersome for some purposes because of the high number of data that is required (i.e., all the points of the segmented shape have to be somehow stored). Therefore, the next problem to be tackled is how to properly *represent* the shape, implying a suitable *shape representation* scheme to be defined with basis on specific tasks. Such schemes may or may not allow the reconstruction of the original shape. In fact, this criterion seems to have been first suggested by [Pavlidis, 1977] with respect to *information preserving* (allow the reconstruction of the original shape) and *information nonpreserving techniques* (do not allow the reconstruction of the original shape). It is worth emphasizing that information preserving representations are particularly important due to the fact that different shapes are mapped onto different representations, whereas nonpreserving techniques can produce equal representations for different shapes (which is called a degenerated or non-inverting mapping, as discussed with respect to functions in Chapter 2). Such nonpreserving techniques are nevertheless usually adopted as shape measures that are useful for shape characterization and classification, as discussed in Chapter 4. Indeed, both approaches, which have their advantages and shortcomings, are frequently applied to shape analysis problems. In addition, it should be observed that some techniques only allow partial reconstruction of the shape.

A more fundamental criterion for characterizing shape representation techniques involves their classification as *boundary-based* and *region-based*. Boundary-based

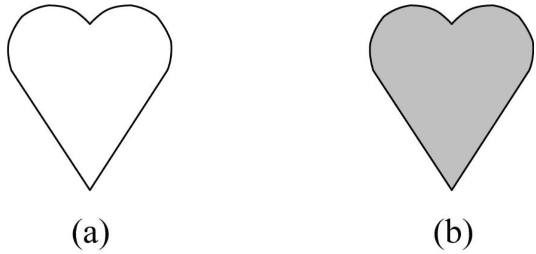


Figure 1.12: *Boundary-based* (a) *and region-based* (b) *shape representations.*

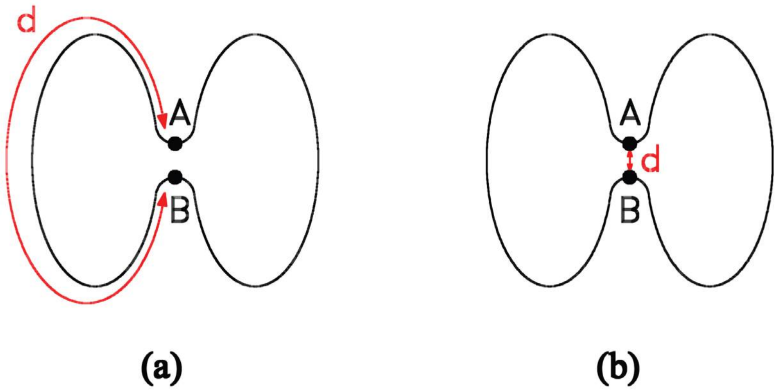


Figure 1.13: *Important difference implied by the boundary-based* (a) *and the region-based* (b) *shape representations.*

(also known as *contour-based*) techniques represent the shape by its outline, while region-based techniques treat the shape in terms of its respective 2D region (see Figure 1.12).

By representing planar regions in terms of one-dimensional signals, contour-based methods allow a simplified one-dimensional approach, while region-based techniques involve 2D signals. This difference frequently implies that contour-based methods are less computationally expensive than region-based methods, though exceptions occur from time to time. Another important difference between these two approaches can be better appreciated through the example in Figure 1.13.

In the contour-based approach, assuming that the contour is traversed counter-clockwise, the distance between the points A and B *along the contour*, indicated as d in Figure 1.13 (a), is larger than the distance in the region-based approach, indicated as d in Figure 1.13 (b). Roughly speaking, if a contour-based local analysis is applied to point A, then point B does not imply a strong influence over the processing performed at A. The opposite situation is verified for the region-based approach, i.e., the point B can significantly affect processing done around A.

Shape Description or Characterization

Some of the most important problems involving shape analysis techniques require extracting information about objects in the real world. For example, one might want to investigate physical properties of biological entities by analyzing their shape, such as when studying spatial coverage, a concept that is also related to shape complexity. Such spatial coverage properties, particularly important in branching structures, can be related to the capacity of an object to interact with its surrounding environment, such as the capacity of roots of trees to extract water and food from the soil or of neural cells to interact with the extracellular medium, including other cells. In situations where relevant shape information is to be extracted, *shape description* or *characterization* techniques have to be applied (description and characterization are used as synonyms in this sense). Moreover, additional and equally important situations where shape description techniques are fundamental arise in shape recognition and shape classification. It should be observed that frequently some shape aspects are more important than others, depending on the task to be solved by the shape analysis system. For instance, many object recognition problems can be solved by first detecting some dominant points that usually occur in the shape (e.g., corners in polygonal figures). Clearly, the type of feature that should be detected depends on each specific problem as well as on the involved shapes, though some features have achieved special importance and popularity in shape analysis. For example, some of the most important aspects of a shape can be detected by analyzing the curvature of the shape boundary, especially in terms of corners and regions of constant curvature, such as circle segments (constant, non-null curvature) or straight lines (constant, null curvature). Furthermore, some shape features are also studied and usually considered as a consequence of biological facts (e.g., psychophysical results indicate that corners play a particularly important role in human shape analysis, see [Attneave, 1954]). To any extent, there are different approaches for extracting information about shapes, which can be classified as follows:

Shape measurements: One of the most common ways of describing shapes involves defining and measuring specific characteristics such as area, perimeter, number of corners, number of holes, curvature-based measures, preferential orientation of contour points, and so on (see Figure 1.14). The underlying idea of the description of a shape by a set of measures (i.e., numbers) is that the obtained measures are sufficient to reasonably represent the relevant information about that shape.

Shape transforms (signal processing-based): Transform techniques are popular in many different areas, from signal processing and telecommunications to

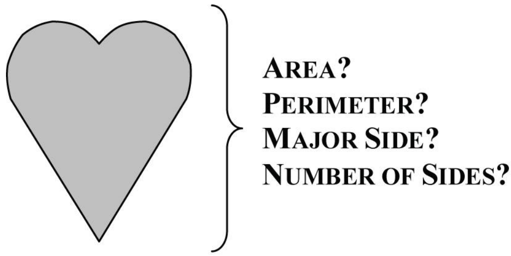


Figure 1.14: *Many important approaches to shape analysis are based on measuring meaningful properties from the shape.*

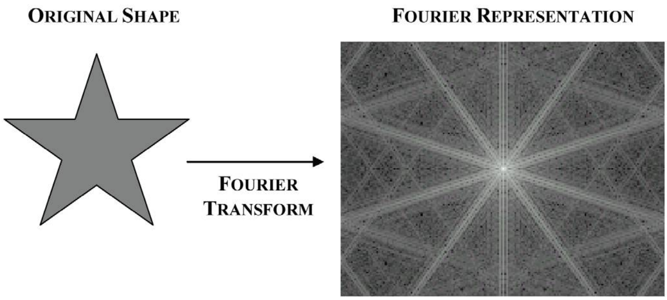


Figure 1.15: *Shape description by using its Fourier transform. Observe that the complete Fourier transform also includes phase information, in addition to the magnitude shown in this figure.*

optics and numerical solution of partial differential equations, also playing an important role in shape analysis. A signal transform is a mathematical tool that expresses the original signal in an alternative way, which is frequently more suitable for a specific task than the original one. For instance, the number two can be alternatively expressed in decimal Arabic (“2”) or in Roman (“II”). While features can be obtained by measuring shape properties directly in their 2D or 3D space, a powerful and widely adopted alternative to such an approach consists in deriving features from transformed shapes. A simple example of obtaining a shape descriptor by using a transform technique is to calculate its Fourier transform (Figure 1.15) and to select some predefined coefficients (e.g., “select the first 5 Fourier coefficients”). There are many different transforms that can be used, though Fourier is one of the most popular. As a matter of fact, the Fourier transform is one of the most powerful and versatile linear transforms. Some other examples of important transforms are the wavelet, the Gabor and the Karhunen-Loève transform. It should be observed that invertible transforms could also be understood

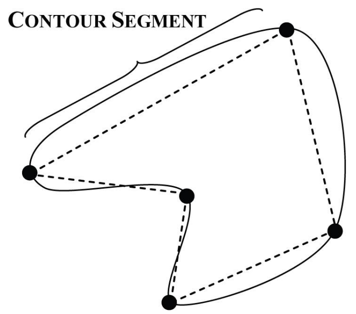
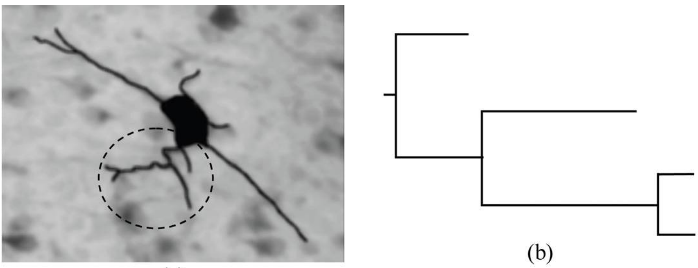


Figure 1.16: *Original shape contour (solid line) and a possible representation by polygonal approximation (dashed line).*

as means for representation of shape. For instance, the original shape can be fully recovered from its Fourier transform representation by using the inverse Fourier transform (see Section 2.7).

Shape decomposition: The third class of shape description techniques presented herein is based on decomposing the shape into simpler parts, which are sometimes called *primitives*, as typically done in the context of structural and syntactical pattern recognition [Fu, 1982]. Since the meaning of "simpler parts" can vary widely in terms of each specific application and types of shapes, knowledge about the problem is usually decisive in this case. Nevertheless, there are some approaches that are versatile enough to be generally considered, being suitable for the most diverse applications. For instance, one of the most important problems in contour analysis involves fitting geometric primitives to contour portions, and the so-called *polygonal approximation* is an excellent example of this approach (Figure 1.16). In the polygonal approximation problem, also known as *piecewise linear approximation*, the original contour must be represented by a set of straight line segments, each line segment representing a portion of the original contour. It is important to note that such a representation can also be used to implement shape processing such as noise filtering (local noisy perturbations occurring in contour portions are eliminated when these portions are represented by line segments) and data compression (e.g., a digital straight contour segment involving hundreds of points can be almost exactly represented in terms of its two extremities). Other examples of shape decompositions are those based on circle segments and 2D polygonal regions, the latter being applied in region-based techniques. Proceeding a step further, the syntactic approach to pattern recognition problems associates abstract symbols to each geometric primitive in such a way that each shape can be represented by a sequence of such symbols. The subsequent shape recognition therefore involves parsing procedures operating over such symbol sequences (or strings; see Chapter 9).

Shape description through data structures: Several problems can be solved by representing aspects underlying the shape in terms of data structures. An illustrative example is the problem of representation of neural dendrites by the so-called dendrograms (typically binary trees), which suit several important applications in neurosciences [Cesar-Jr. & Costa, 1999; Costa et al., 2000] (see Figure 1.17). In addition to presenting a clear representation of



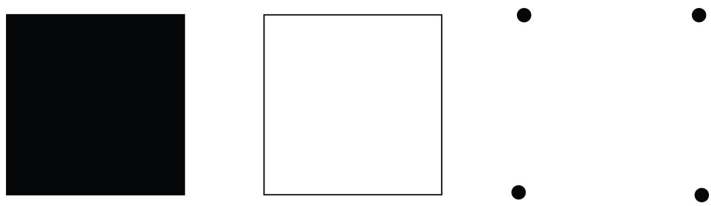
(a) (b)

Figure 1.17: *A neural cell is presented in* (a) *while a dendrogram of one of its dendrites (that with terminations and branch points indicated) is shown in* (b).

the branching pattern, such hierarchical data structures can incorporate important additional shape measures such as size, width, local bending energy and angles in a particularly compact way. As a matter of fact, dendrograms have become important in neuroscience because they can be easily stored and handled by computer programs, thus allowing the standardization required for exchanging data among different laboratories, scientists and other professionals. It should be borne in mind that dendrograms are not only important in neuroscience, but also provide valuable descriptions of virtually any other branched structure, such as rivers, trees, vascular systems, etc.

Shape Visualization

Scientific visualization techniques are mainly concerned with the suitable presentation of large amounts of data to humans. As such, this area is particularly important both for supporting the development of shape analysis tools and as an aid for shape inspection by human operators. In the former situation, shape visualization can be used to effectively present the obtained results (e.g., features to be tested, intermediate results, filtered images), which can involve the superposition of such results over the original shapes or relating the several obtained data, in order to provide insights about the assets and shortcomings of the considered techniques. On the other hand, shape visualization is also important to aid human experts to solve specific problems, e.g., to help a physician decide how a broken bone should be treated.



(a) (b) (c)

Figure 1.18: *An example of shape compression: the shape in* (a) *can be represented by its outline* (b)*, which can then be represented by its corners* (c)*. Therefore, the shape in* (a)*, which could require hundreds or thousands of points in order to be properly represented in a discrete space, can now be conveniently represented by only the 4 points shown in* (c)*.*

Shape Compression

Digital image applications generally involve processing a large amount of data, which can become prohibitive depending on the application, especially when real-time processing is required. Data compression is an issue often present in imaging applications, including shape analysis problems. For instance, applications that depend on large image databases (e.g., fingerprint recognition) usually require storing and computing very large sets of images. Some shape analysis approaches naturally offer good data compression solutions, e.g., contour-based approaches, which represent 2D shapes by 1D structures (Figure 1.18). Very high compression rates can be obtained by further compressing such contours. In fact, there are some approaches for image and video coding (for data compression) which make extensive use of contour shape representations (e.g., [Buhan et al., 1997]).

1.3.3 Shape Classification

Finally, after shape processing, representation and characterization (often involving feature extraction), classification algorithms are usually applied in order to assign each considered shape to a category. There are two particularly important aspects related to shape classification. The first is the problem of, given an input shape, deciding whether it belongs to some specific predefined class. This can also be thought of as a shape recognition problem, usually known as *supervised classification*. The second equally important aspect of shape classification is how to define or identify the involved classes in a population of previously unclassified shapes. This represents a difficult task, and expert knowledge acquisition problems are usually involved. The latter situation is known as *unsupervised classification* or *clustering*. Both supervised and unsupervised classification involve comparing shapes, i.e., deciding how *similar* two shapes are, which is done, in many situations, by *matching* specially important corresponding points of them (typically landmarks or

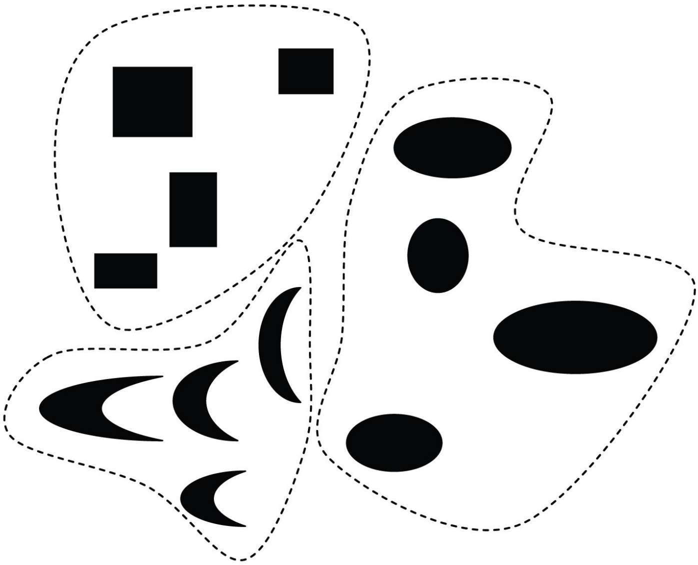


Figure 1.19: *In an unsupervised shape classification problem, the algorithm should discover shape classes from a given set of unclassified shapes.*

saliences). These four topics are outlined in the following sections. General shape classification algorithms are covered in Chapter 8.

Unsupervised Shape Classification

As further explained in Chapter 8, classifying a shape can be understood as the problem of assigning some class to it. Nevertheless, in many cases defining the shape classes is itself a difficult problem. Therefore, it is important to devise methods that attempt to find shape classes based only on the unclassified pattern data, an approach that is commonly known as *unsupervised learning*. The identification of data clusters in the data sets is an ordinary way of defining shape classes, which is carried out by *clustering algorithms*. For instance, for a given set of geometrical shapes, such as those shown in Figure 1.19, the expected output of a clustering algorithm would be the three sets indicated by the dashed lines in that figure.

Supervised Shape Classification

When the shape classes are predefined, or examples are available for each class, it is often desirable to create algorithms that take a shape as input and assign it to one of the classes, i.e., that it *recognizes* the input shape (see Figure 1.20). For instance, an important problem in medical imaging involves the recognition of mammographic calcifications in order to verify the presence of tumors, the shapes of which are

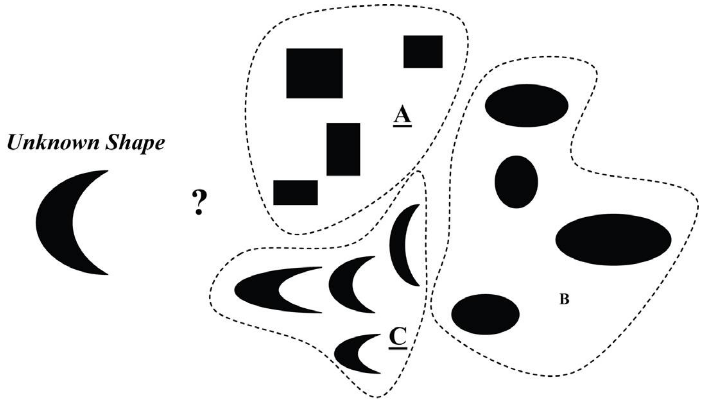


Figure 1.20: *Supervised shape classification: given a set of shape classes A, B and C, and an unknown shape, to which class does the unknown shape belong?*

related to the tumors being malignant or not. Observe that the terms *shape recognition* and *supervised shape classification* are often used interchangeably.

Shape Similarity

Shape similarity refers to establishing criteria that allow objective measures of how much two shapes are similar (or different) to each other, including issues such as when a given shape A can be considered more similar to another shape B than to C. An example is shown in Figure 1.21. It is worth observing that shape similarity criteria, which are fundamental to classifying shapes, are generally dependent on each specific problem. For instance, in a situation where size is an important parameter, two shapes with similar areas can be more similar to each other than two shapes with significantly different areas. Clearly, the shape features adopted for their characterization play a central role with respect to defining how similar two shapes are. Shape similarity is particularly important when trying to match two or more shapes.

Shape Matching

Shape matching is the process through which two or more shapes are associated, generally in a point-by-point fashion. There are many different applications for such techniques. For instance, images of the same region of the human body can be obtained using different acquisition modalities, such as tomography and magnetic resonance, and an important task in such situations is to *register* or *align* each image, in order to create a correspondence map between the several representations (see Figure 1.22). This task is a particular example of a more general problem known as *image registration*. One approach that can solve this problem

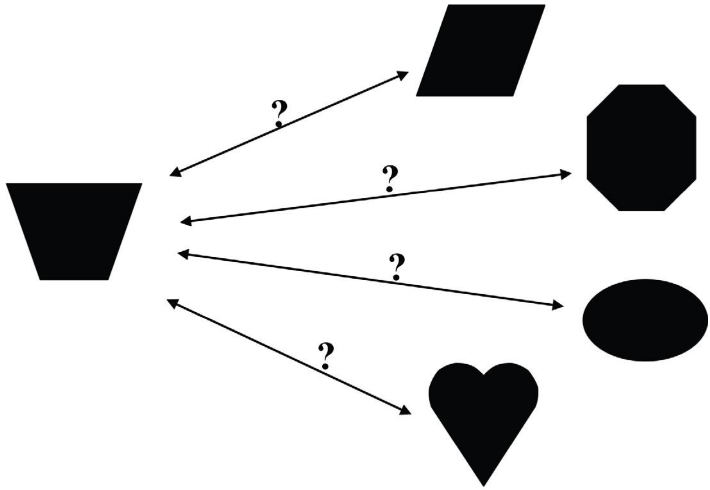


Figure 1.21: *Shape similarity: which shape is more similar? How can similarity be objectively measured?*

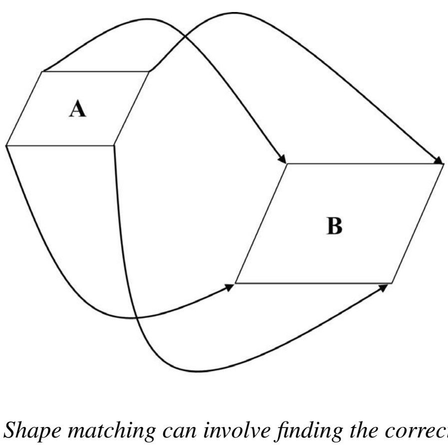


Figure 1.22: *Shape matching can involve finding the correct corresponding points between a given shape A and a target shape B.*

involves the detection of instances of homologous structures in both images. In addition, shape matching is important to different problems in data fusion and 3D reconstruction. In the former application, information about the same object is obtained by using different sensors, and the respective representations have to be merged. On the other hand, the latter involves establishing a correspondence between two-dimensional shapes obtained as slices of a three-dimensional structure: this correspondence allows reconstructing an approximation of the original 3D object. The reader interested in sensor fusion, image registration and shape matching is referred to [Bloch & Maître, 1997; Burr, 1981; Davis, 1979; Milios, 1989; Viola & Wells, 1997].

1.4 Additional Material

The book now includes a special box section called *Additional material*, which includes links to useful online material. In particular, the authors keep some open-source software projects online which are directly related to the theory and methods introduced in the book. The reader is invited to visit the projects' homepages, use the software and help to develop it in a open-source collaboration. The main projects, which are also referred in appropriated places in the book, are listed as an example in the box below.

Additional resources: ***Slides, videos, software***

- Interactive image segmentation: `http://segmentacao.incubadora.fapesp.br/portal`
- Vessel segmentation: `http://retina.incubadora.fapesp.br/portal`
- Dimensionality reduction: `http://dimreduction.incubadora.fapesp.br/portal`
- Shape analysis: `http://code.google.com/p/imagerecognitionsystem/`

Also, slides to help in courses based on the book are now available at the book homepage: `http://www.ime.usp.br/~cesar/shape/`
