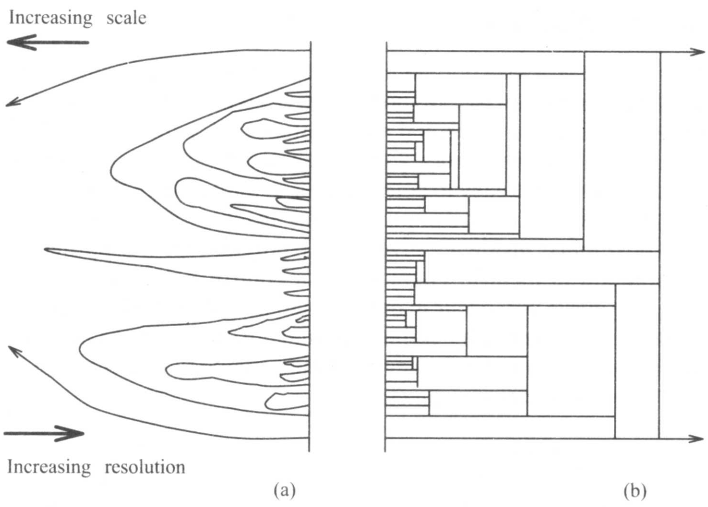

Figure 8.15: Scale-space image. (a) Varying number and locations of curve segmentation points as a function of scale. (b) Curve representation by an interval tree.

are achieved at different resolutions. This problem is no less important if a curve is to be divided into segments; some curve segmentation points exist in one resolution and disappear in others without any direct correspondence. Considering this, a **scale-space** approach to curve segmentation that guarantees a continuously changing position of segmentation points is a significant achievement [Babaud et al., 1986; Witkin, 1986; Yuille and Poggio, 1986; Maragos, 1989; Florack et al., 1992; Griffin et al., 1992]. In this approach, only new segmentation points can appear at higher resolutions, and no existing segmentation points can disappear. This is in agreement with our understanding of varying resolutions; finer details can be detected in higher resolution, but significant details should not disappear if the resolution increases. This technique is based on application of a unique Gaussian smoothing kernel to a one-dimensional signal (e.g., a curvature function) over a range of sizes and the result is differentiated twice. To determine the peaks of curvature, the zero-crossing of the second derivative is detected; the positions of zero-crossings give the positions of curve segmentation points. Different locations of segmentation points are obtained at varying resolution (different Gaussian kernel size). An important property of the Gaussian kernel is that the location of segmentation points changes continuously with resolution which can be seen in the **scale-space image** of the curve, Figure 8.15a. Fine details of the curve disappear in pairs with increasing size of the Gaussian smoothing kernel, and two segmentation points always merge to form a closed contour, showing that any segmentation point existing in coarse resolution must also exist in finer resolution. Moreover, the position of a segmentation point is most accurate in finest resolution, and this position can be traced from coarse to fine resolution using the scale-space image. A multi-scale curve description can be represented by an **interval tree**, Figure 8.15b. Each pair of zero-crossings is represented by a rectangle, its position corresponding with segmentation point locations on the curve, its height showing the lowest resolution at which the segmentation point can be detected. Interval trees can be used for curve decomposition in different scales, keeping the possibility of segment description using higher-resolution features.

Another scale-space approach to curve decomposition is the **curvature primal sketch** [Asada and Brady, 1986] (compare Section 11.1.1). A set of primitive curvature discontinuities is defined and convolved with first and second derivatives of a Gaussian in multiple resolutions. The curvature primal sketch is computed by matching the multi-scale convolutions of a shape. The curvature primal sketch then serves as a shape representation; shape reconstruction may be based on polygons or splines. Another multi-scale border-primitive detection technique that aggregates curve primitives at one scale into curve primitives at a coarser scale is described in [Saund, 1990]. A robust approach to multi-scale curve corner detection that uses additional information extracted from corner behavior in the whole multi-resolution pyramid is given in [Fermuller and Kropatsch, 1992].

8.2.5 B-spline representation

Representation of curves using piecewise polynomial interpolation to obtain smooth curves is widely used in computer graphics. B-splines are piecewise polynomial curves whose shape is closely related to their control polygon—a chain of vertices giving a polygonal representation of a curve. B-splines of the third order are most common because this is the lowest order which includes the change of curvature. Splines have very good representation properties and are easy to compute: First, they change their shape less then their control polygon, and they do not oscillate between sampling points as many other representations do. Furthermore, a spline curve is always positioned inside a convex $n + 1$-polygon for a B-spline of the n^{th} order—Figure 8.16. Second, the interpolation is local in character. If a control polygon vertex changes its position, a resulting change of the spline curve will occur in only a small neighborhood of that vertex. Third, methods of matching region boundaries represented by splines to image data are based on a direct search of original image data. These methods are similar to the segmentation methods described in Section 6.2.6. A spline direction can be derived directly from its parameters.

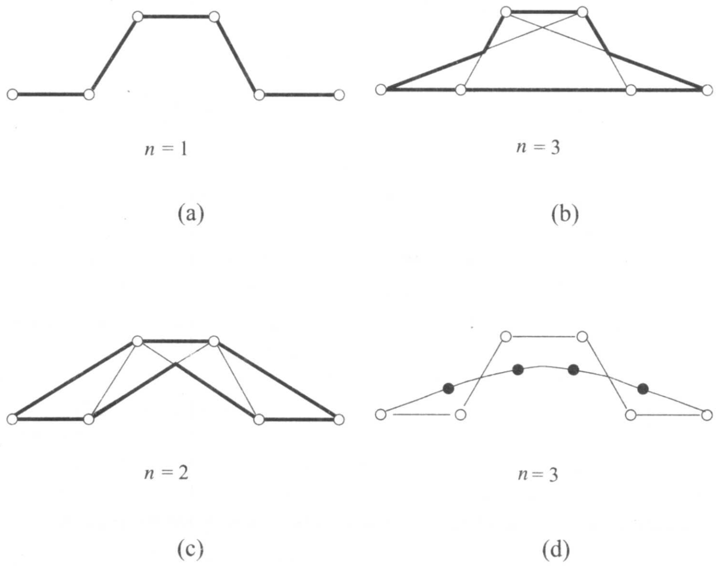

Figure 8.16: Splines of order n. (a), (b), (c) Convex $n + 1$-polygon for a B-spline of the n^{th} order. (d) 3^{rd}-order spline.

Let $\mathbf{x}_i$, $i = 1, \ldots, n$ be points of a B-spline interpolation curve; call this interpolation curve $\mathbf{x}(s)$. The s parameter changes linearly between points $\mathbf{x}_i$—that is, $\mathbf{x}_i = \mathbf{x}(i)$. Each part of a cubic B-spline curve is a third-order polynomial, meaning that it and its first and second derivatives are continuous. B-splines are given by

$$\mathbf{x}(s) = \sum_{i=0}^{n+1} \mathbf{v}_i \, B_i(s), \tag{8.17}$$

where $\mathbf{v}_i$ are coefficients representing a spline curve, and $B_i(s)$ are base functions whose shape is given by the spline order. The coefficients $\mathbf{v}_i$ bear information dual to information about the spline curve points $\mathbf{x}_i$—the values $\mathbf{v}_i$ can be derived from $\mathbf{x}_i$ values and vice versa. The coefficients $\mathbf{v}_i$ represent vertices of the control polygon, and if there are n points $\mathbf{x}_i$, there must be $n + 2$ points $\mathbf{v}_i$. The two end points $\mathbf{v}_0$, $\mathbf{v}_{n+1}$ are specified by binding conditions. If the curvature of a B-spline curvature is to be zero at the curve beginning and end, then

$$\mathbf{v}_0 = 2\,\mathbf{v}_1 - \mathbf{v}_2 \,,$$
$$\mathbf{v}_{n+1} = 2\,\mathbf{v}_n - \mathbf{v}_{n-1} \,. \tag{8.18}$$

is closed, then $\mathbf{v}_0 = \mathbf{v}_n$ and $\mathbf{v}_{n+1} = \mathbf{v}_1$.

The base functions are non-negative and are of local importance only. Each base function $B_i(s)$ is non-zero only for $s \in (i-2, i+2)$, meaning that for any $s \in (i, i+1)$, there are only four non-zero base functions for any i: $B_{i-1}(s)$, $B_i(s)$, $B_{i+1}(s)$, and $B_{i+2}(s)$. If the distance between the $\mathbf{x}_i$ points is constant (e.g., unit distances), all the base functions are of the same form and consist of four parts $C_j(t), j = 0, \ldots, 3$.

$$C_0(t) = \frac{t^3}{6},$$

$$C_1(t) = \frac{-3t^3 + 3t^2 + 3t + 1}{6}$$

$$C_2(t) = \frac{3t^3 + 6t^2 + 4}{6}$$

$$C_3(t) = \frac{-t^3 + 3t^2 + 3t + 1}{6}$$

Because of equation (8.17) and zero-equal base functions for $s \notin (i-2, i+2)$, $\mathbf{x}(s)$ can be computed from the addition of only four terms for any s

$$\mathbf{x}(s) = C_{i-1,3}(s)\,\mathbf{v}_{i-1} + C_{i,2}(s)\,\mathbf{v}_i + C_{i+1,1}(s)\,\mathbf{v}_{i+1} + C_{i+2,0}(s)\,\mathbf{v}_{i+2}\,. \tag{8.19}$$

Here, $C_{i,j}(s)$ means that we use the j^{th} part of the base function B_i (see Figure 8.17). Note that

$$C_{i,j}(s) = C_j(s-i), \qquad i = 0, \ldots, n+1, \qquad j = 0, 1, 2, 3. \tag{8.20}$$

To work with values inside the interval $[i, i+1)$, the interpolation curve $\mathbf{x}(s)$ can be computed as

$$\mathbf{x}(s) = C_3(s-i)\,\mathbf{v}_{i-1} + C_2(s-i)\,\mathbf{v}_i + C_1(s-i)\,\mathbf{v}_{i+1} + C_0\,\mathbf{v}_{i+2}\,. \tag{8.21}$$

Specifically, if $s = 5$, s is positioned at the beginning of the interval $[i, i+1)$, therefore $i = 5$ and

$$\mathbf{x}(5) = C_3(0)\,\mathbf{v}_4 + C_2(0)\,\mathbf{v}_5 + C_1(0)\,\mathbf{v}_6 = \mathbf{v}_4 + \mathbf{v}_5 + \mathbf{v}_6\,, \tag{8.22}$$

or, if $s = 7.7$, then $i = 7$ and

$$\mathbf{x}(5) = C_3(0.7)\,\mathbf{v}_6 + C_2(0.7)\,\mathbf{v}_7 + C_1(0.7)\,\mathbf{v}_8 + C_0(0.7)\,\mathbf{v}_9\,. \tag{8.23}$$

Other useful formulae can be found in [DeBoor, 1978; Ballard and Brown, 1982].

Splines generate curves which are usually considered pleasing. They allow a good curve approximation, and can easily be used for image analysis curve representation problems. A technique transforming curve samples to B-spline control polygon vertices is described in [Paglieroni and Jain, 1988] together with a method of efficient computation of boundary curvature, shape moments, and projections from control polygon vertices. Splines differ in their complexity; one of the simplest applies the B-spline formula for curve modeling as well as for curve extraction from image data [DeBoor, 1978]. Splines are used in computer vision to form exact and flexible inner model representations of complex shapes which are necessary in model-driven segmentation and in complex image understanding tasks. On the other hand, splines are highly sensitive to change in scale.

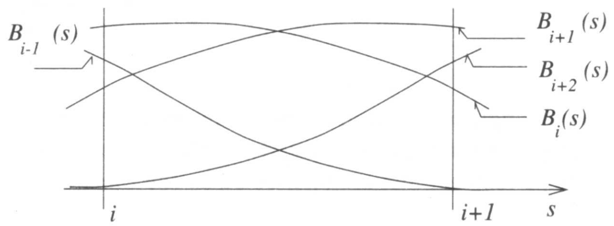

Figure 8.17: The only four non-zero base functions for $s \in (i, i+1)$.

8.2.6 Other contour-based shape description approaches

Many other methods and approaches can be used to describe two-dimensional curves and contours.

The **Hough transform** has excellent shape description abilities and is discussed in detail in the image segmentation context in Section 6.2.6 (see also [McKenzie and Protheroe, 1990]). Region-based shape description using **statistical moments** is covered in Section 8.3.2 where a technique of contour-based moments computation from region borders is also included. Further, it is necessary to mention the **fractal** approach to shape [Mandelbrot, 1982], which is also used for shape description.

Mathematical morphology can be used for shape description, typically in connection with region skeleton construction (see Section 8.3.4) [Reinhardt and Higgins, 1996]. A different approach is introduced in [Loui et al., 1990], where a **geometrical correlation function** represents two-dimensional continuous or discrete curves. This function is translation, rotation, and scale invariant and may be used to compute basic geometrical properties.

Neural networks (Section 9.3) can be used to recognize shapes in raw boundary representations directly. Contour sequences of noiseless reference shapes are used for training, and noisy data are used in later training stages to increase robustness; effective representations of closed planar shapes result [Gupta et al., 1990]. Another neural network shape representation system uses a modified Walsh-Hadamard transform (Section 3.2.2) to achieve position-invariant shape representation.

8.2.7 Shape invariants

Shape invariants represent a very active current research area in machine vision. Although the importance of shape invariance has been known for a long time, the first machine vision-related paper about shape invariants appeared in [Weiss, 1988], followed by a book [Kanatani, 1990]. The following section gives a brief overview of this topic and is based mostly on a paper [Forsyth et al., 1991] and on a book [Mundy and Zisserman, 1992] in which additional details can be found. The book [Mundy and Zisserman, 1992] gives an overview of this topic in its Introduction, and its Appendix presents an excellent and detailed survey of projective geometry for machine vision. Even if shape invariance is a novel approach in machine vision, invariant theory is not new, and many of its principles were introduced in the nineteenth century.

As has been mentioned many times, object description is necessary for object recognition. Unfortunately, all the shape descriptors discussed so far depend on viewpoint, meaning that object recognition may often be impossible as a result of changed object or observer position, as illustrated in Figure 8.18. The role of shape description invariance is obvious—shape invariants represent properties of such geometric configurations which remain unchanged under an appropriate class of transforms. Machine vision is especially concerned with the class of projective transforms.

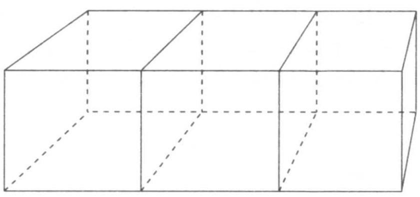

Figure 8.18: Change of shape caused by a projective transform. The same rectangular cross section is represented by different polygons in the image plane.

Collinearity is the simplest example of a projectively invariant image feature. Any straight line is projected as a straight line under any projective transform. Similarly, the basic idea of the projection-invariant shape description is to find such shape features that are unaffected by the transform between the object and the image plane.

A standard technique of projection-invariant description is to hypothesize the pose (position and orientation) of an object and transform this object into a specific co-ordinate system; then shape