Curvature Morphology

F. Leymarie and M. D. Levine
TR-CIM-88-26
Center for Intelligent Machines
McGill University, Montreal, Canada

December 1988*

Abstract

The notion of curvature of planar curves has emerged as one of the most powerful for the representation and interpretation of objects in an image. Although curvature extraction from a digitized object contour would seem to be a rather simple task, few methods exist that are at the same time easy to implement, fast, and reliable in the presence of noise. In this paper we first briefly present a scheme for obtaining the discrete curvature function of planar contours based on the chain-code representation of a boundary. Secondly, we propose a method for extracting important features from the curvature function such as extrema or peaks, and segments of constant curvature. We use mathematical morphological operations on functions to achieve this. Finally, on the basis of these morphological operations, we suggest a new scale-space representation for curvature named the *Morphological Curvature Scale-Space*. Advantages over the usual scale-space approaches are shown.

Index terms: Shape Representation, Curvature Representation and Analysis, Morphological Operators, Multiscale Description, Multi-Dimensional Scale-Space.

*2nd printing by F.Leymarie, released in August 2001, while at Brown University.

Sommaire

Le concept de courbure appliqué aux courbes et contours planaires est reconnu comme un outil des plus utiles pour la représentation et l'interprétation de projections d'objets dans une image numérique. L'obtention de la fonction de courbure à partir d'un contour d'objet peut sembler à première vue simple. Pourtant, très peu de méthodes existent qui soient à la fois aisément programmable, rapide en temps de calculs et robuste en présence de bruit ou de perturbations. Dans le présent article, nous présentons, dans un premier temps, une telle méthode, afin d'obtenir une fonction de courbure discrète sous une forme adéquate, ce sur la base de la représentation dîte du code de chaîne ("chain-code"). Par la suite, nous proposons une méthode pour extraire et identifier les caractéristiques morphologiques de la fonction de courbure, telles que les extrema et segments à courbure constante ou plateaux. Pour ce faire nous utilisons des opérateurs définis dans le cadre de la Morphologie Mathématique. Finalement, sur la base de ces opérateurs morphologiques, nous introduisons une nouvelle représentation multi-échelles de la fonction discrète de courbure. Les avantages de cette représentation sont démontrés.[1]

[1]The authors would like to thank B. Kimia, A. Dobbins, G. Godin and J. Lagarde of the McGill Research Center for Intelligent Machines (McRCIM) for useful discussions related to the work on curvature, P. Noble and D. Gauthier for discussions related to the cell shape problem, and M. Dalziel for proof reading. F. Leymarie wishes to thank P. Cohen of *Ecole Polytechnique de Montreal* for introducing him to mathematical morphology and the Natural Science and Engineering Research Council (NSERC) for a postgraduate scholarship to support his studies. This research was partially supported by the Medical Research Council of Canada, Grant No. MT-3236. M. D. Levine would like to thank the Canadian Institute for Advanced Research for its support.

Contents

1 Introduction

Shape representation and analysis is fundamental to computer vision and our interest in it is an outgrowth of a study of the dynamic changes in cell shape [28, 21]. Of the many approaches to shape that have been proposed, the notion of curvature of planar curves has emerged as one of the most powerful for the representation and interpretation of objects in an image [18]. Curvature is a measure of the rate of change in orientation at each point along a curve. There is psychophysical, physiological, as well as computational and mathematical support in favor of using curvature as a representation for contours. Curvature extrema seem to be used by the human visual system to segment contours into meaningful parts [2, 5, 16]. For example, endstopped neurons in the visual cortex can be interpreted as performing local curvature measurements [12]. Local estimations of curvature and tangent information are sufficient for the recovery of the trace of a curve in an image [31]. From differential geometry, the fundamental theorem of the local theory of curves states that any regular[2] planar curve is uniquely defined by its curvature [11]. Non-regular points of a curve are singular points where the curvature goes to infinity, that is, where the change in orientation is undefined. They correspond to visually salient points such as corners. Therefore, given the set of singular points of a contour, which can be represented as extrema of curvature, as well as the curvature values for all intervening points, a contour is uniquely defined. This representation by curvature is invariant to rigid motion, that is, with respect to translation or rotation.

In a typical computer vision system, discrete contours of objects are first extracted from an image. Curvature of these discretized contours is then approximated and used to detect important features of the boundary of an object. Although curvature extraction from an object contour would seem to be a rather simple task, few methods exist that are simultaneously easy to implement, fast and reliable in the presence of noise. In this communication we first briefly propose a scheme for obtaining the discrete curvature function of planar contours (opened or closed) based on the chain-code representation of a boundary [14]. This approach has been previously reported and we mention only our own attempts to optimize its implementation.

Secondly, we present a way of extracting important features from the curvature function. We are interested in localizing features such as extrema or peaks of curvature, points of inflection (*i.e.*, zero-crossings of curvature), and segments of constant curvature that correspond to straight line segments or circular arcs of the boundary. Furthermore, we would like to be able to differentiate these features by their relative significance, that is, by the degree of their isolation from nearby features, their relative amplitude, and their scale or size. Consequently, we seek methods for segmenting the curvature function into its basic and significant events. To achieve this goal we propose using morphological operations on functions [36]. These operators permit us to create a representation of curvature, not only in terms of feature localization and identification, but also in terms of significance and scale. We will demonstrate that morphological operators can be used to remove details of a signal, in our case the curvature function. This can be done for a continuum of sizes without

[2]Regularity implies continuity of a curve and its derivatives.

blurring the shape of these details and while still retaining the global shape features. This property of morphological operations permits us to suggest a new scale-space representation for curvature referred to as the *Morphological Curvature Scale-Space*. Advantages over the usual scale-space approaches [40, 1, 27, 23] will become evident from the discussions. In these references, scale is a function of the amount of blurring or smoothing necessary to remove and deform details of a signal.

In the following sections we first present how to retrieve the orientation and curvature from the discrete trace of a contour using as a basis the chain-code representation. In this context, quantization errors and "protrusion-depression definition" problems inherent to the chain-code representation are discussed. We also indicate how smoothing of discrete orientation data should be performed to extract its derivative and curvature, all the while reducing the noise amplification as much as possible. Then we present the kind of morphological operations that are applicable to curvature analysis. Certain morphological measures are also introduced for the purpose of describing curvature peaks extracted using these morphological operations. The Morphological Curvature Scale-Space is then defined and its main advantages demonstrated.

2 From Discrete Contour to "Not Too Smooth" Curvature

Since any visual input to a computer is discrete[3] as a result of the digitization mesh, contour extraction must address the issue of synthesizing a continuous representation from a discrete one [22]. The problem involves going from the discrete trace of image contours (*i.e.*, edge pixels) to a continuous or linked set of points through which the curve passes. Essentially, two kinds of encoding and representation schemes of smooth curves have been proposed in the computational vision literature: curve fitting algorithms [33, 32, 38, 8] and orientation chaining algorithms [14, 24, 15, 6, 1, 29, 26].

In the first scheme, a set of curves is fit to the contour by minimizing a certain error measure. The points where curves meet are retained as knot points and are essentially arbitrary. Most such techniques employ polygonal approximations, circular arcs, or splines. Their main drawback is that the knot points are not always related to our perception of the salient points of a contour, such as for example the curvature extrema. Furthermore, the a priori assumption made about the nature of contour segments between knot points, that is, that they are composed of straight lines, circular arcs, or splines, can often be seen to be rather rigid.

In the second scheme, an opposite approach is adopted where orientation or curvature along the contour is first approximated and then processed further to extract the knot points. These points are usually obtained by examining the rate of change of orientation (curvature function) along the contour. Curve segments can then be fitted between the knot points. Since this representation seems to be supported at the physiological level and is also advantageous at the computational level, we have chosen it to encode and represent

[3] This is also true of the human visual system and of its discrete retinal sampling grid.

discrete contours.[4]

2.1 Curvature Function Extraction

Two different approaches have been reported for computing curvature of a discrete curve from orientation. In the first type of method (hereafter called FD methods, for Filtering and Differentiation), smoothed local curvature is evaluated by convolving a discrete orientation representation of the contour[5] with a template. The derivative of a Gaussian of variable size is often employed as the template in order to extract an approximation of the curvature [1]. Alternatively, the orientation data can be differentiated by computing the first difference of the angle formed by nearby points and then smoothed with a Gaussian filter to remove noise. This can yield a multiscale representation if a set of different sized filters is used [10]. However, this second method usually has a poorer signal-to-noise ratio. This can be attributed to a specific reason: filtering discrete orientation data with a template that directly extracts the derivative greatly amplifies noise.

In the second type of method (hereafter called DOS methods, for Difference Of Slopes), curvature is estimated at each point by taking the angular difference between slopes (orientation) of two line segments fitted to the data before and after each point [35, 29]. This can be repeated for line segments of varying length. The extent of data smoothing is governed in both methods by the size variable: the template size for FD's and the line size for DOS's.

Although DOS methods give good results in the presence of boundary noise, they do require more expensive computations than FD methods. They involve fitting straight lines to the data at each contour pixel. In this paper, we will propose an alternative to extracting curvature from discrete orientation data which is also an FD method, but which requires even less complex computations . In the next section, we describe how the discrete orientation representation is obtained.

2.2 From the Trace of the Discrete Contour to a Discrete Orientation Representation

Let us consider the discrete trace of a curve[6] on a square sampling lattice with integer coordinates and denote it by $v(u) = (x(u), y(u))$, where u represents the distance between the centroids of two neighboring pixels. On a square sampling lattice, u takes the value 1 for direct neighbors (*i.e.,* either vertical or horizontal) or the value $\sqrt{2}$ for indirect or diagonal neighbors. Given a discrete trace of a curve such as $v(u)$, a common first step is to consider

[4] Some recent methods proposed in the literature can be seen as a compromise between the two abovementioned schemes. For example, Medioni and Yasumoto process cubic B-splines fitted to a contour by examining their curvature. Extrema of curvature are then extracted and the fitting process is refined [25].

[5] The discrete orientation representation is usually given by the well-known chain code representation of Freeman [14]. This representation is considered in more detail in the following section.

[6] The discrete trace is usually obtained in a previous segmentation step. For example, an edge map may be derived from an image and from it an object contour is defined by linking pixels.

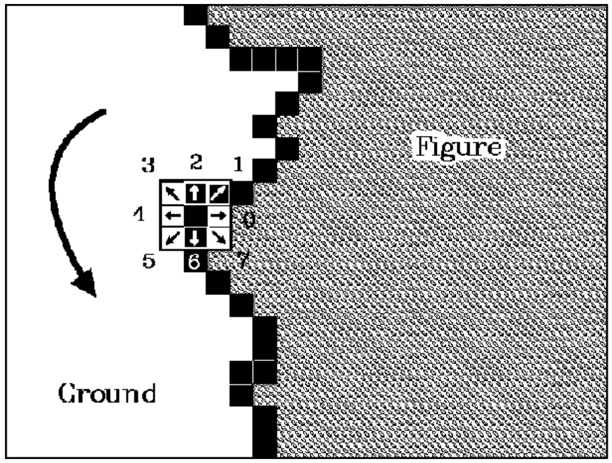


Figure 1: Example of chain code generation.

the chain code [14] of the bounding contour. We observe that the chain code is in fact a discrete representation of the orientation along the contour. It can be stated as a vector of integers $C(u)$, where each entry represents the discrete angle formed by adjacent pixels. Due to the nature of the square sampling lattice, angles in the chain code are constrained to a discrete range of eight values (0, ..., 7) representing multiples of 45^o. Contour pixels are traversed in a fixed counterclockwise direction. Therefore, the figure or interior of the object is defined as being to the left of the traversal direction (see Figure 1).

Two quantization problems inherent to the chain code definition must be solved before embarking on further processing of the data [24]. First, the angle quantization in such a narrow range (0,...,7) may introduce angle discontinuities of more than 180^o (*e.g.*, where a 0 follows a 7). The solution here is to avoid discontinuities by using a modulo 8 operation that puts a bound of π on angle discontinuities. For example, replace 0 by 8, if 0 follows 7. This modified chain code $C_m(u)$ is generated as follows:

- $C_m(0) = C(0)$,
- If $|C_m(i-1) - C(i)| > 4$ then find an integer k_i such that $|C_m(i-1) - [C(i) + 8 * K_i]| \leq 4$, set $C_m(i) = C(i) + 8 * K_i$,
- Otherwise, $C_m(i) = C(i)$.

The second quantization error is introduced by the discretization factor u (discrete arc length unit) which is usually simply incremented by one for each new pixel found while traversing of the contour. However, a diagonal step is actually $\sqrt{2}$ times longer than a step along the grid axes. This can be corrected by counting two arc length units for entries of $C_m(u)$ with directions along grid axes and three arc length units for entries along diagonal directions (approximates $\sqrt{2}$ by 1.5 while keeping integers values). We denote by s this normalization of the distance u along the trace of the curve.

In addition to these quantization errors, there is a third source of error which is related to the definition of the chain code. Very sharp protrusions ending in a line of pixels of width one (see Figure 2) could be erroneously defined as a depression, depending on the preceding values of C_m as one gets closer to the tip of the protrusion. Therefore, protrusion definition problems occur at angles of $\pm 180^o$ or equivalently, where $|C_m(i-1) - C(i)| = 4$. The same problem can also occur for depressions, depending on how the trace of the contour is obtained.[7] In the general case, a protrusion or depression with an angle of $\pm 180^o$ that occurs at a contour pixel $P(i)$ can be checked by examining a neighborhood of $P(i)$ to ensure that the interior of the object is still defined to be on the left of the direction of contour traversal. A simple procedure consists of looking at contour pixels preceding and following $P(i)$. A description of the procedure follows:

- If $|C_m(i-1) - C(i)| = 4$, then
  - Compare the coordinates of the contour pixels $P(i-k)$ and $P(i+k)$ adjacent to $P(i)$ until two are found which are located at different spatial positions.
  - Check whether or not $P(i+k)$ is to the *left* of $P(i-k)$. If it is, then we have a protrusion. That is, $C_m(i) = C_m(i-1) + 4$. Otherwise, $C_m(i) = C_m(i-1) - 4$, indicating a depression.

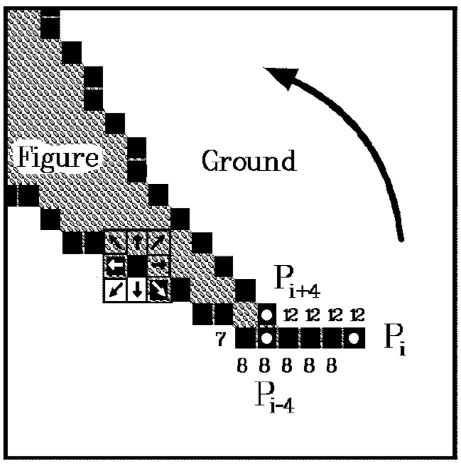
In summary, the original chain code $C(u)$ must be corrected for three types of errors: discontinuities of more than $|180^o|$, arc length normalization, and protrusion-depression definition problems. These three types of errors can be detected in parallel by applying the above-mentioned procedures. The steps which follow address the problem of obtaining smoother orientation data and then deriving the curvature.

2.3 From Discrete Orientation to Smoothed Curvature Representation

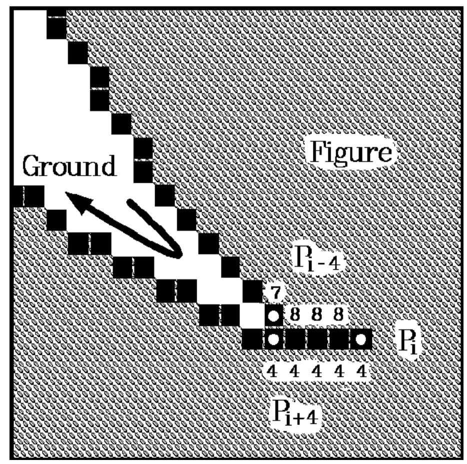
If we define $\theta(s)$ as the modified chain code after both quantization and protrusion-depression errors have been corrected, we now must process it to obtain the curvature measurements. By definition, for a parameterization of the curve by arc length s, the curvature $k(s)$ is given by $k(s) = \theta'(s)$.

Noise amplification problems occur when differentiating a discrete signal such as $\theta(s)$ because it is very coarsely quantized in 45^o steps. For example, a straight line at an angle between 0^o and 45^o is represented by a succession of 0^o and 45^o orientations, a very noisy signal indeed. This aliasing phenomenon cannot be alleviated unless the original sampling grid is refined or the orientation sampling is done over a larger neighborhood. Still, the effect of aliasing can be reduced by the application of low-pass or band-pass filtering to the signal $\theta(s)$. Therefore, it is common to filter $\theta(s)$ with a Gaussian template G_σ [6, 10], with standard deviation σ, defined as:

[7] A simple object contour following algorithm used to extract the chain code from a segmented edge map would not usually tolerate, as part of the contour, depressions made of a string of pixels of width one. A more "intelligent" contour following algorithm would extract contour pixels in two passes. It would turn around the object (interior on the left) as well as "around" the background (exterior on the left) thereby generating sharp protrusions as well as sharp depressions.



(a) Protrusion.



(b) Depression.

Figure 2: Protrusion-depression definition problems. A protrusion is shown in (a) where an angle of $\pm 180^o$ occurs at the contour pixel P_i. The right choice for the chain code value is 12. If the usual chain code definition were used, a value of 4 would be obtained, leading to an inconsistent definition of the tip as being part of a depression, as shown in (b). Using our proposed procedure, since pixel $P(i+k)$ lies to the left of pixel $P(i-k)$, a protrusion is recognized. In (b), since pixel $P(i+k)$ lies to the right of pixel $P(i-k)$, a depression is recognized.

$$G_\sigma(s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{\frac{-s^2}{2\sigma^2}\}\,. \tag{1}$$

Furthermore, the convolution theorem [30] can be used to differentiate the filtered signal:

$$(\theta(s) \star G_\sigma)' = \theta(s) \star G'_\sigma\,. \tag{2}$$

This is the approach usually followed to obtain $k(s)$, whereby the smoothing and differentiation of $\theta(s)$ with a band-pass filter G'_σ is performed at the same time [6, 1]. However, in terms of computational complexity, there is an advantage to using Gaussian templates in an initial smoothing step instead of computing the derivatives of the Gaussian directly, as is done in other FD methods. Gaussian filtering can be very efficiently implemented by combining *Hierarchical Discrete Correlation* (HDC) techniques [7] with *cascaded convolution* [9, 39, 19]. Burt's HDC algorithm permits very fast computation for Gaussian smoothing at variable scale, while the cascaded convolution property can be combined with HDC techniques to obtain a broader range of scaling (smoothing) than by solely employing HDC techniques. Reusing the convolution theorem and the cascaded convolution property of Gaussian templates we obtain a smoothed curvature signal k_σ as follows:

$$k_\sigma(s) = (\theta(s) \star G_\sigma)' = (\theta(s) \star G_{\sigma_1} \star G_{\sigma_2})' = \theta(s) \star G_{\sigma_1} \star G'_{\sigma_2}\,, \tag{3}$$

where σ_1 and σ_2 are the standard deviations for the two Gaussian functions G_{σ_1} and G_{σ_2}, respectively, and

$$\sigma^2 = \sigma_1^2 + \sigma_2^2\,. \tag{4}$$

Therefore, in the first step we filter $\theta(s)$ with a low-pass filter using an HDC-like algorithm and denote this operation by:

$$\theta_{\sigma_1}(s) = \theta(s) \star G_{\sigma_1}\,. \tag{5}$$

Then, in a second step, we filter $\theta_{\sigma_1}(s)$ with a band-pass filter to obtain its smoothed derivative:

$$k_\sigma(s) = \theta_{\sigma_1}(s) \star G'_{\sigma_2}\,. \tag{6}$$

The first step involving G_{σ_1} is performed very quickly due to the HDC implementation. The second step is performed for a derivative of a Gaussian template G'_{σ_2} over a much smaller neighborhood than for G'_σ in equation (2). Our complete procedure for extracting curvature from discrete orientation data is illustrated in Figure 3.[8]

Although Gaussian filtering is necessary to reduce noise effects and to evaluate the derivative of $\theta(s)$ with a numerically stable procedure, this smoothing process must be performed in a conservative way. By "conservative smoothing" we imply that we wish to retain as far as possible the relevant details of $\theta_{\sigma_1}(s)$ and $k_\sigma(s)$. However, Gaussian filtering

[8]The object shown in Figure 3 will be used throughout this paper to illustrate the complete curvature morphology approach.

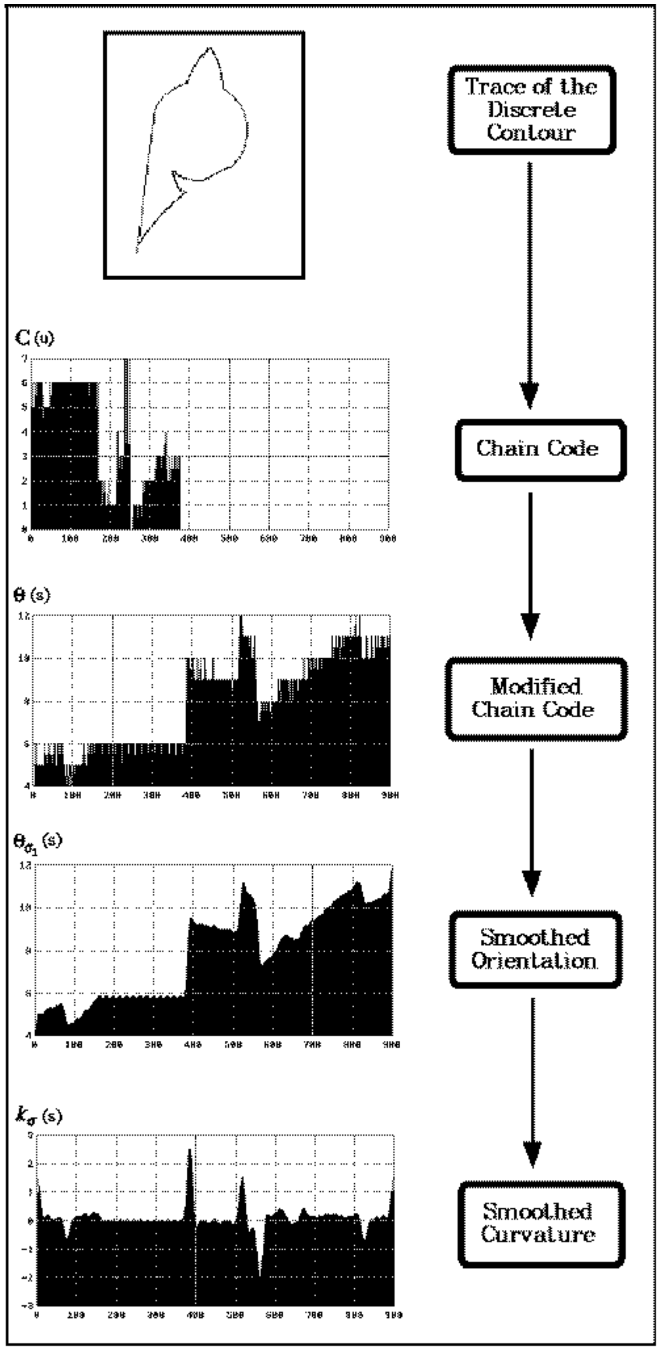


Figure 3: From the trace of the discrete contour to smoothed orientation and curvature representations.

considers noise and peaks of $k_\sigma(s)$ without distinguishing between them on the basis of size [29]. We would also like to describe the features of $k_\sigma(s)$ by detecting them without having to significantly modify $k_\sigma(s)$. Gaussian filtering cannot be used for this second purpose since peaks corresponding to features would be attenuated and blurred. In summary, linear filtering using Gaussian templates and their derivatives is ineffective because the scope of the filtering is too global, that is, indiscriminate with respect to significant curvature features [4, 34, 29].

We seek a description of curvature features in terms of their annihilation, as proposed in the so-called *scale-space* approaches [40, 6, 17]. However, instead of using Gaussian smoothing to generate a scale-space, which has an undesirable global effect, we propose the use of morphological operators [36]. These permit the removal of details from a signal such as $k_\sigma(s)$ without modifying its global "morphology" or shape. Since they exert only local influence, morphological operators have the desirable property of generating uniform scale-spaces that are unambiguous and therefore easily interpreted. Morphology for curvature is the subject of the next section.

3 Curvature Analysis

Although the curvature at each point along a curve uniquely defines its behavior, it is the morphology of curvature that permits the retrieval of useful visual information. We would like a description of $k_\sigma(s)$ in terms of its singularities and any well identified regions or arcs.

As mentioned earlier, peaks in curvature are known to correspond to the critical points of a contour and are useful for visual perception, as well as mathematical and computational representation. Constant regions are defined by straight lines ($k_\sigma(s) = 0$) and arcs of constant curvature ($k_\sigma(s) =$ constant). They can be used to compress information and reduce the complexity of the interpretation process. Such constant regions, along with their associated peaks of curvature, can also be used to compute region-based representations of an object. Examples are the so-called skeleton (symmetric-axis transform) [10, 21] and local rotational symmetries [13]. Zero crossings of $k_\sigma(s)$ map to inflection points of a contour. Since they are a natural way of segmenting the contour into convex and concave regions, we hereafter separate $k_\sigma(s)$ into two functions, $k_+(s)$ for convex regions and $k_-(s)$ for concave regions (see Figure 4).

In addition to localizing the curvature features, we wish to be able to quantify their significance. This includes their relative importance (*e.g.*, relative amplitude of peaks), as well as their extent and isolation from one another along the contour. This concept of feature significance leads naturally to a multiscale representation in which irrelevant details of $k_\sigma(s)$ can be removed before its interpretation.

All of these notions relating to feature shape, localization, size, isolation and scale are associated and can be treated using the mathematical technique known as *mathematical morphology* [36].

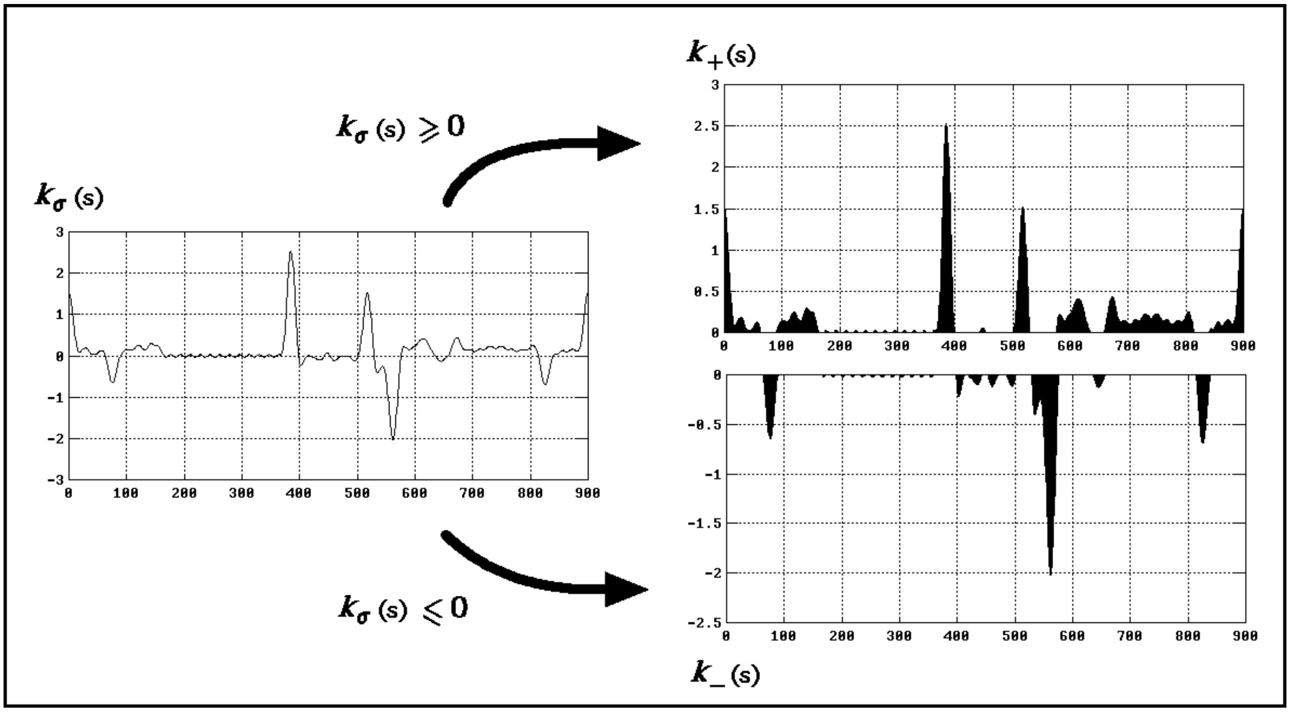


Figure 4: Curvature function $k_\sigma(s)$ as the sum of two functions, $k_+(s)$ and $k_-(s)$.

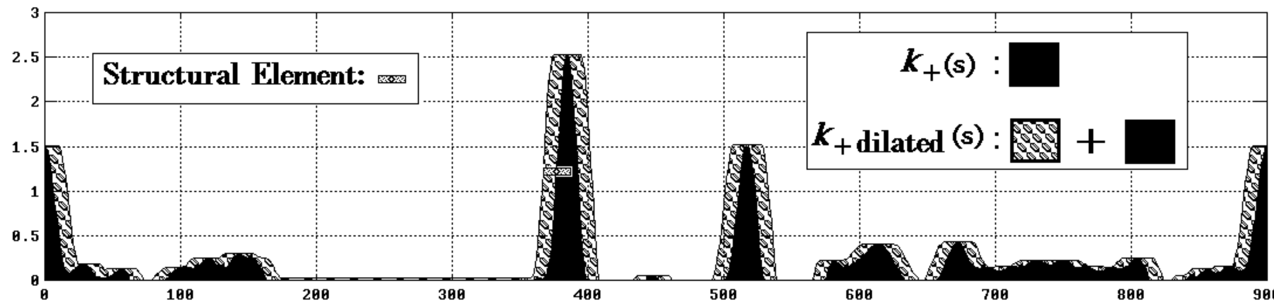
3.1 Curvature Morphology

Mathematical morphology applied to functions such as $k_\sigma(s)$ provides us with useful tools for the extraction of primitives or dominant shapes found in such functions.[9] Two dual operations, *erosion* and *dilation*, are the keystones of this approach. As their names indicate, erosion is a shrinking operation while dilation is an expanding one. These operations are performed locally by observing the structure of the neighborhood at each point of the function. The neighborhood over which local computations are performed is defined by a *structural element* defined by a set of resolution cells constituting a specific shape such as a line or square.[10] In our application to curvature, only flat structural elements, that is, lines of increasing width are considered.[11] These are sufficient for the extraction of peaks and flat regions of $k_\sigma(s)$. Eroding a function by a segment of size R is equivalent to replacing the function values at every point by the minimum of all the points in a neighborhood of radius R. Likewise, dilating a function by a segment of size R is equivalent to a maximum transformation over a neighborhood of radius R. Examples of dilation and erosion applied

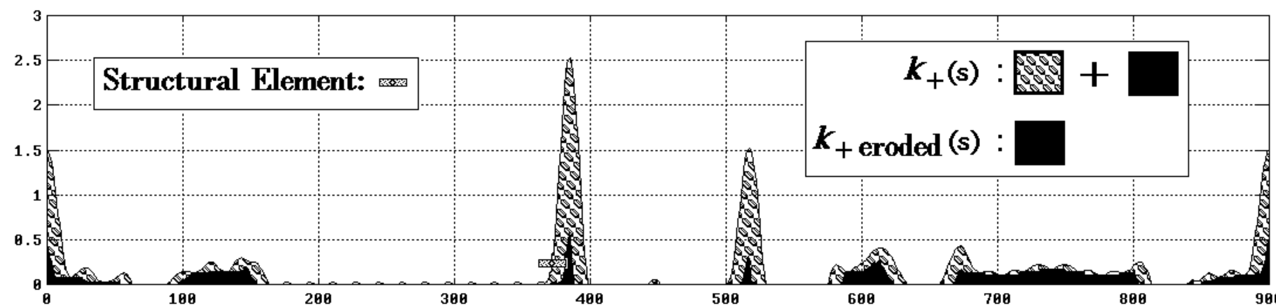
[9]See Appendix A for mathematical definitions and properties of morphological operations on functions. For further mathematical details about this subject, the reader is referred to chapter twelve of Serra's book [36]. For applications to grey level images, see the paper by Sternberg on grayscale morphology [37].

[10]This is analogous to the concept of a two-dimensional window that we can slide over an image to perform convolution-like computations.

[11]We employ *symmetric* flat structural elements centered on a function point in the examples given throughout this paper. Therefore, we use only widths of odd values for these structural elements. Furthermore, the minimal width is fixed to three arc length units (see §B).



(a) Dilation operation.



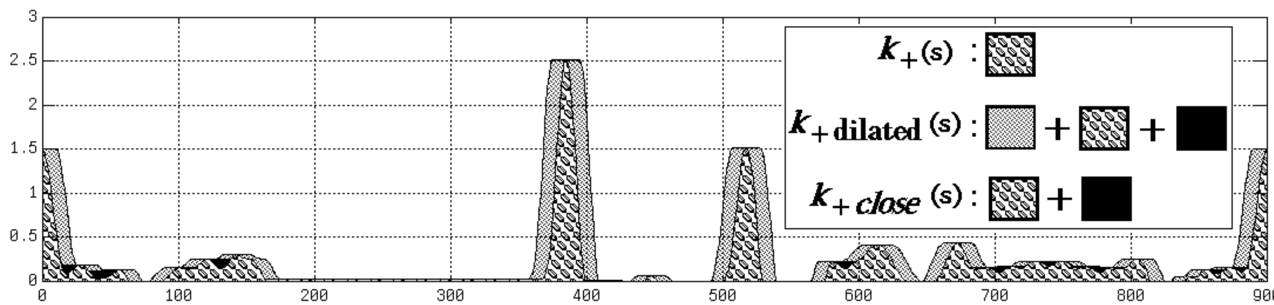
(b) Erosion operation.

Figure 5: Examples of dilation and erosion applied to $k_+(s)$ (positive curvature function of object shown in Figure 3). The structural element shown in (a) and (b) defines the neighborhood over which local min-max computations are performed (here 21 arc length units wide). In (a), a dilation is performed by using a maximum operation. In (b), an erosion is performed by using a minimum operation.

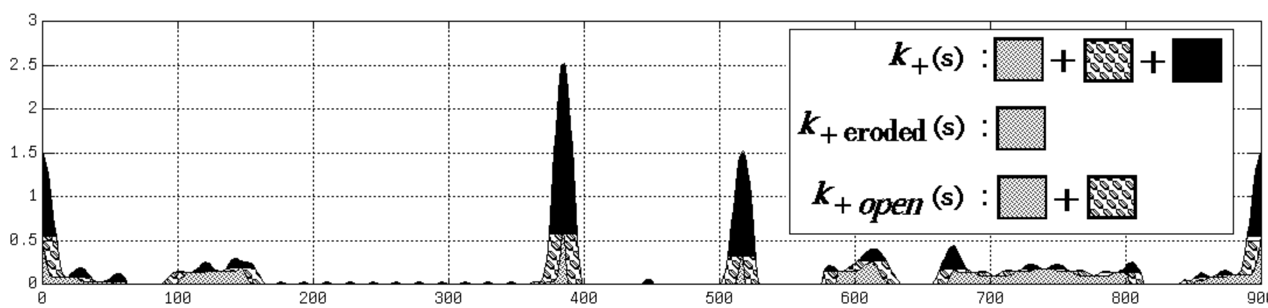
to $k_+(s)$ are given in Figure 5 (similar results are obtained for $k_-(s)$).

By combining dilation and erosion two new operations can be defined: *opening* and *closing*. Opening is the dilation of an eroded function, while closing is the erosion of a dilated function. In both cases, by combining the two dual operations of dilation and erosion, the original function is only partially recovered since some details are segmented from it. In the case of opening, convexities or "bumps" of increasing size are removed with the use of different sized structural elements. Closing, on the other hand, is used to fill-in concavities or holes. These concepts are shown in Figure 6. A dilation (erosion) will remove from the top (bottom) of the function all details that are smaller than the structural element size. The result is a new function which is more regular (smoother) than the original one.

We observe that mathematical morphology provides us with a method for removing from the signal $k_\sigma(s)$ details of increasing size. This is indicated in Figure 7.(a) where



(a) Closing operation.



(b) Opening operation.

Figure 6: Examples of closing and opening operations on a positive curvature function for a structural element of fixed width (21 arc length units wide).

we compare the function $k_+(s)$ to its "opened" version $k_{+open}(s)$ and thereby isolate the peaks and bumps. Thus the residual $k_+(s) - k_{+open}(s)$ is defined as the *top-hat transform* of $k_+(s)$. We can also define its dual, the *bottom-hat transform* in which $k_+(s)$ is compared to its "closed" version $k_{+close}(s)$ according to the residual $k_+(s) - k_{+close}(s)$. These two residuals are similarly defined for the negative curvature function as $k_-(s) - k_{-open}(s)$ and $k_-(s) - k_{-close}(s)$ respectively. The bottom-hat transform applied to $k_-(s)$ is illustrated in Figure 7.(b). Both these operators will be used as the basic morphological operations for curvature. Hat-transforms are so named because they can be visualized as a covering of peaks with a hat of fixed size.

3.2 Peak Description

Peaks with bases of different size can be extracted by varying the size of the flat structural element. Thus, by increasing the width of the structural element details of greater and greater extent can be extracted from $k_+(s)$ (similarly for $k_-(s)$). Once a peak has been isolated from the curvature signal, its morphology or shape can be analyzed. We define four morphological measures to describe an isolated peak:

1. The *extent* of the peak, δ_s which is equal to the width of the flat structural element.

2. The *maximal* peak *amplitude* or maximal relative curvature, k_{max}.

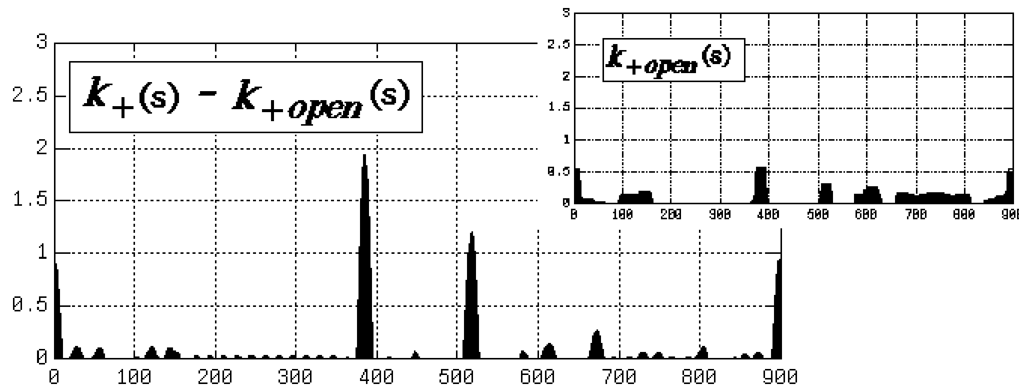
3. The *average* relative *curvature* k_{avg}, given by the area under the curve defined by the peak (Σk) divided by δ_s.

4. The *shape factor* r, given by the ratio of k_{max} to k_{avg} ($r \geq 1$).

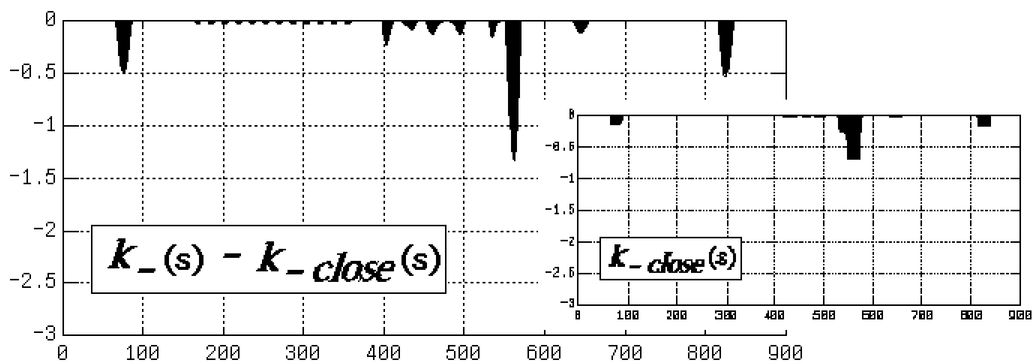
The peak features k_{max}, δ_s and Σk are illustrated in Figure 8. Each of the four morphological measures derived from them can be used to describe the nature of the peak. Peak significance can be determined not only by large values of k_{max} with respect to δ_s but also by the shape factor r. Values of r greater than one may indicate a very narrow and well-defined peak. On the other hand, a value close to one indicates a region of constant curvature (see Figure 9.(a)). By examining its extent δ_s, this constant curvature region could be retained as being significant or be disregarded. The different possible kinds of shapes of peaks described by the shape factor r are illustrated in Figure 9. Stable curvature peaks are represented by the range $1 < r \leq 2$ (see Figure 9.(b) and (c)). By "stable" we mean that if δ_s is increased, k_{max} of the given peak will substantially increase. On the other hand, when $2 < r < \delta_s$, the peak tends to flatten for increasing δ_s (see Figure 9.(d)). The case of $r \approx \delta_s$ corresponds to noise in the curvature data (see Figure 9.(e)).

An example of peak description using the four morphological measures for peaks extracted using the top-hat transform is given in Figure 10.

Although these shape measures seem to be useful, the scale or size at which they should be sought is variable and unknown *a priori*. Therefore, curvature morphology analysis should be performed at different scales, where scale is defined as the variable size of the structural element. This leads to a multiscale representation of curvature which is developed in the next section.



(a) Top-hat transform.



(b) Bottom-hat transform.

Figure 7: Examples of hat-transforms for a structural element of fixed width (21 arc length units wide). In (a) the top-hat transform is applied to $k_+(s)$ to extract its peaks. In the top-right hand corner is shown the effect of removing the extracted peaks from $k_+(s)$. This is equivalent to $k_{+open}(s)$. In (b) the bottom-hat transform is applied to $k_-(s)$ to extract its valleys. In the bottom-right hand corner is shown the effect of removing the extracted valleys from $k_-(s)$. This is equivalent to $k_{-close}(s)$.

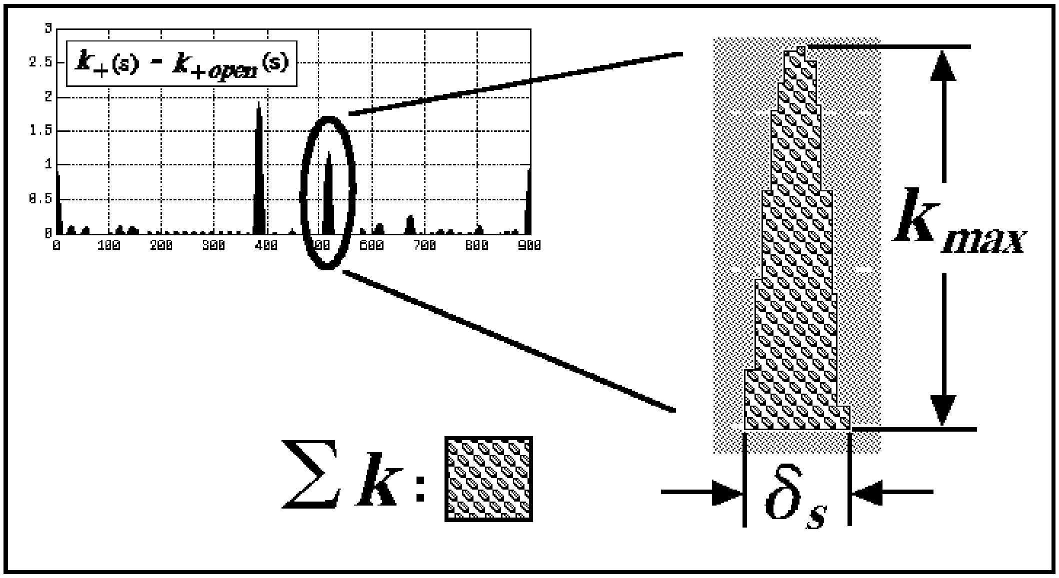


Figure 8: The features used to describe an isolated peak of curvature.

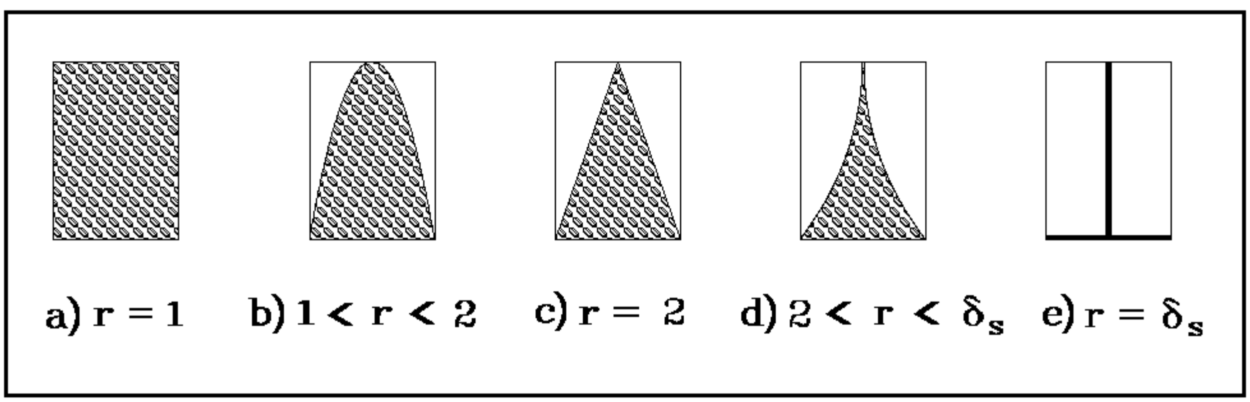
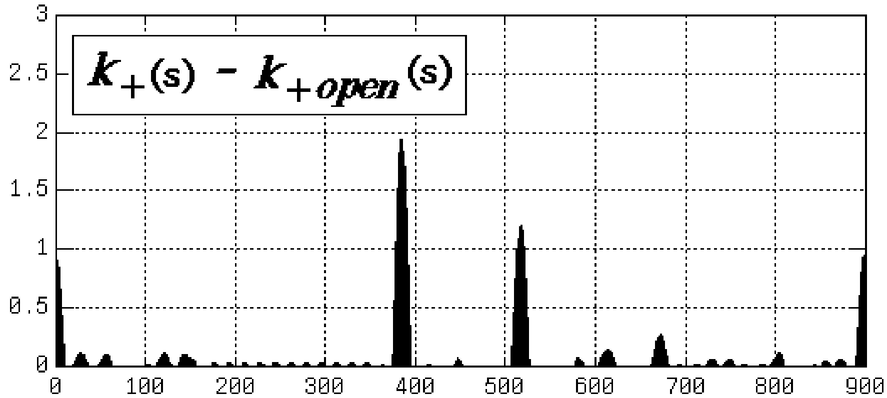


Figure 9: The five possible types of peak shapes described by the shape factor r for given k_{max} and δ_s. In (a) $k_{avg} = k_{max}$. In (b), (c) and (d) are shown the three most common types of curvature peaks. In (e) $\Sigma k \approx k_{max}$.



No. of peaks = 35, disc mid-width = 10

Peak#	δ_s	Smax	$\sum$ K	Kavg	Kmax	r
0	21	899	11.7326	0.5587	0.9573	1.7134
1	19	27	1.1781	0.0620	0.1081	1.7430
2	21	55	0.9555	0.0455	0.0997	2.1917
3	11	100	0.0798	0.0073	0.0164	2.2577
4	21	119	1.0946	0.0521	0.1072	2.0570
5	21	142	1.3328	0.0635	0.1045	1.6471
6	9	175	0.1581	0.0176	0.0291	1.6559
7	9	192	0.1581	0.0176	0.0291	1.6559
8	9	209	0.1581	0.0176	0.0291	1.6559
9	9	226	0.1581	0.0176	0.0291	1.6559
10	9	243	0.1581	0.0176	0.0291	1.6559
11	9	260	0.1581	0.0176	0.0291	1.6559
12	9	277	0.1581	0.0176	0.0291	1.6559
13	9	294	0.1581	0.0176	0.0291	1.6559
14	9	311	0.1581	0.0176	0.0291	1.6559
15	9	328	0.1581	0.0176	0.0291	1.6559
16	9	345	0.1581	0.0176	0.0291	1.6559
17	5	362	0.0133	0.0027	0.0047	1.7813
18	21	384	24.5310	1.1681	1.9414	1.6619
19	4	414	0.0187	0.0047	0.0076	1.6189
20	11	446	0.3975	0.0361	0.0619	1.7124
21	21	516	15.1986	0.7237	1.2043	1.6641
22	12	580	0.4545	0.0379	0.0699	1.8461
23	21	613	1.7960	0.0855	0.1358	1.5876
24	21	672	3.1908	0.1519	0.2679	1.7633
25	10	693	0.0727	0.0073	0.0156	2.1433
26	9	712	0.0883	0.0098	0.0180	1.8314
27	17	729	0.5219	0.0307	0.0528	1.7195
28	17	748	0.5219	0.0307	0.0528	1.7195
29	9	765	0.0883	0.0098	0.0180	1.8314
30	11	784	0.0801	0.0073	0.0164	2.2499
31	19	803	1.1237	0.0591	0.1166	1.9722
32	5	843	0.0151	0.0030	0.0057	1.8759
33	12	855	0.2728	0.0227	0.0437	1.9226
34	15	872	0.4743	0.0316	0.0573	1.8124

Figure 10: Examples of peak description using the four morphological measures: δ_s, k_{max}, k_{avg} and r. The structural element size is fixed at 21 arc length units as before. S_{max} stands for the position of k_{max} in arc length units. Σk values are given for the sake of completeness.

3.3 Morphological Curvature Scale-Space

We generate the multiscale representation of $k_+(s)$ (similarly for $k_-(s)$) by uniformly increasing the size of the structural element. A sequence of peaks is generated using the hat-transforms. A scale history can then be associated with each peak, starting with the scale at which it appears and terminating with the scale at which it ceases to increase in height. As soon as a peak stops increasing it no longer needs to be considered as part of $k_+(s)$. Once a peak has been removed from $k_+(s)$, no other peak is permitted to grow over the extent of the former. This is simply because only one curvature feature can be associated with a given contour segment. An example of the scale-space generated in this way is given in Figure 11. Here, the position of an event, that is of a peak of $k_+(s)$, is given by the position S_{max} of the maximal relative curvature value k_{max} associated with the peak.

Our scale-space representation for curvature (hereafter referred to as the Morphological Curvature Scale-Space or MCS) possesses certain desirable advantages over previous approaches such as the "Curvature Primal Sketch" [1]. For example, each branch in the scale-space diagram remains isolated whereas in the usual scale-space approaches reported in the literature instabilities occur because of branch merges [40, 1, 27]. The MCS bears some similarity to the kind of scale-space generated under the *weak-continuity* constraints scheme of Blake and Zisserman [4, 3], although in their representation, only discontinuities in curvature are tracked explicitly. Furthermore, the MCS satisfies the three criteria proposed by Perona and Malik [34], that any candidate paradigm for generating multiscale representations should satisfy.[12] These criteria are:

Causality: No spurious details should be generated when the scale (δ_s) is increased.

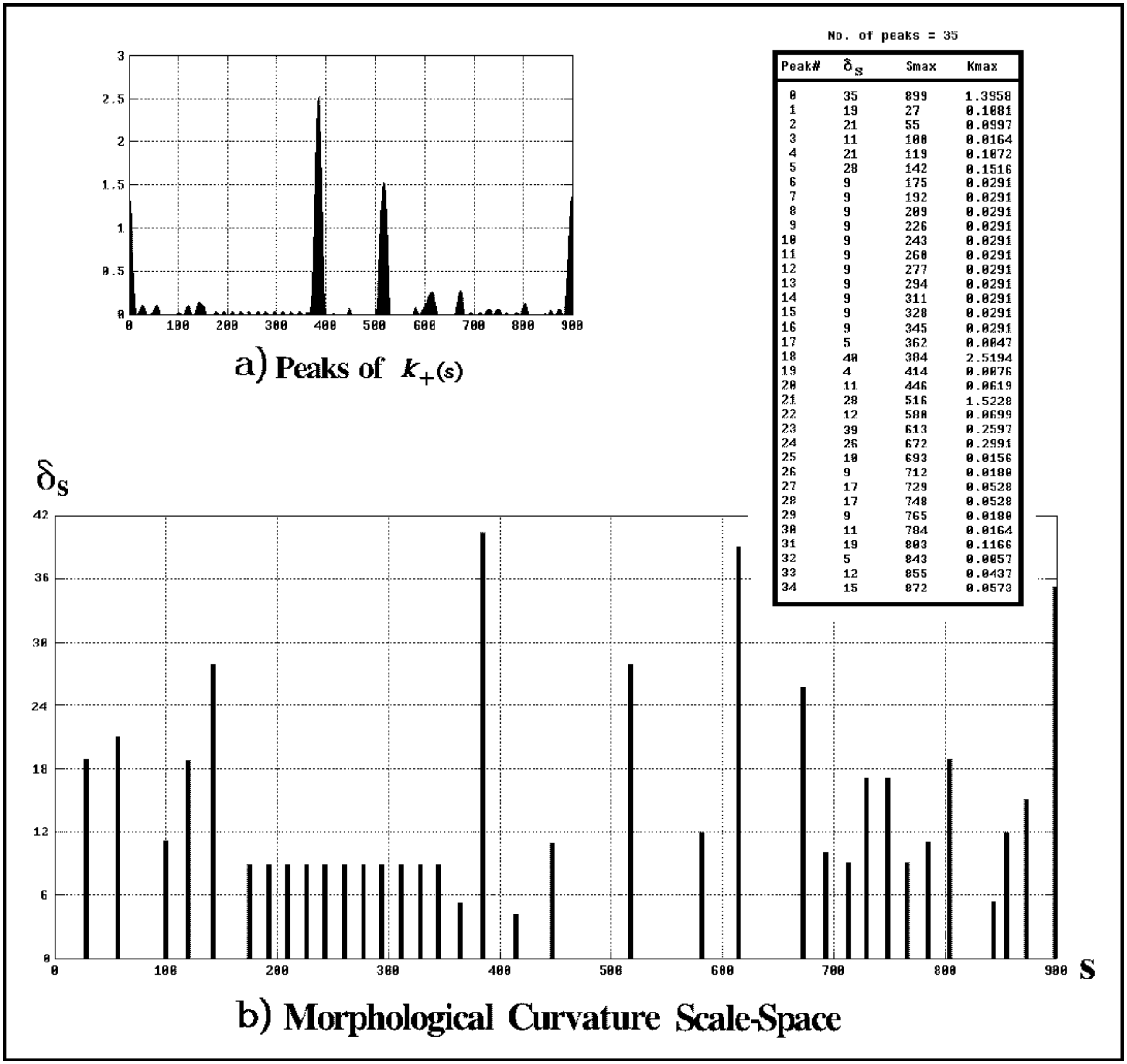
Immediate Localization: At each scale, feature boundaries should be sharp (unblurred) and correspond to meaningful boundaries at that scale.

Piecewise Smoothing: At all scales it is preferable that intra-feature smoothing occur rather than inter-feature smoothing.

The MCS is an elegant way of removing noise from $k_+(s)$ while *preserving* its significant features. Noisy elements of $k_+(s)$ which exhibit a short history at small scales in the MCS can be eliminated. Note that both the top-hat and the down-hat transforms can be used here to remove small convex bumps and to fill-in small concave depression of $k_+(s)$, respectively. An example of filtering $k_+(s)$ in this fashion is shown in Figure 12, where the threshold on δ_s for removing bumps and valleys was set at 12 in Figure 11. A second threshold, this time on the minimal acceptable $k_{max}(s)$ value, is also used for peak significance. This threshold was set at 0.20 in Figure 11. The same filtering process is also applied to $k_-(s)$. The combined results are shown in Figure 13. This ability of the MCS to deal with noise justifies our previous conservative attitude toward Gaussian smoothing (see § 2.3).

The MCS can also be used to build a hierarchy of significant peaks. Peaks with long histories starting at a small scale are classified as prominent. Peaks starting at higher scales,

[12]Scale-space representations based on the usual Gaussian blurring approach only satisfy the causality criterion.



Peak#	δ_s	Smax	Kmax
0	35	899	1.3958
1	19	27	0.1081
2	21	55	0.0997
3	11	100	0.0164
4	21	119	0.1072
5	28	142	0.1516
6	9	175	0.0291
7	9	192	0.0291
8	9	209	0.0291
9	9	226	0.0291
10	9	243	0.0291
11	9	260	0.0291
12	9	277	0.0291
13	9	294	0.0291
14	9	311	0.0291
15	9	328	0.0291
16	9	345	0.0291
17	5	362	0.0047
18	40	384	2.5194
19	4	414	0.0076
20	11	446	0.0619
21	28	516	1.5228
22	12	580	0.0699
23	39	613	0.2597
24	26	672	0.2991
25	10	693	0.0156
26	9	712	0.0180
27	17	729	0.0528
28	17	748	0.0528
29	9	765	0.0180
30	11	784	0.0164
31	19	803	0.1166
32	5	843	0.0057
33	12	855	0.0437
34	15	872	0.0573

Figure 11: Example of the morphological curvature scale-space.

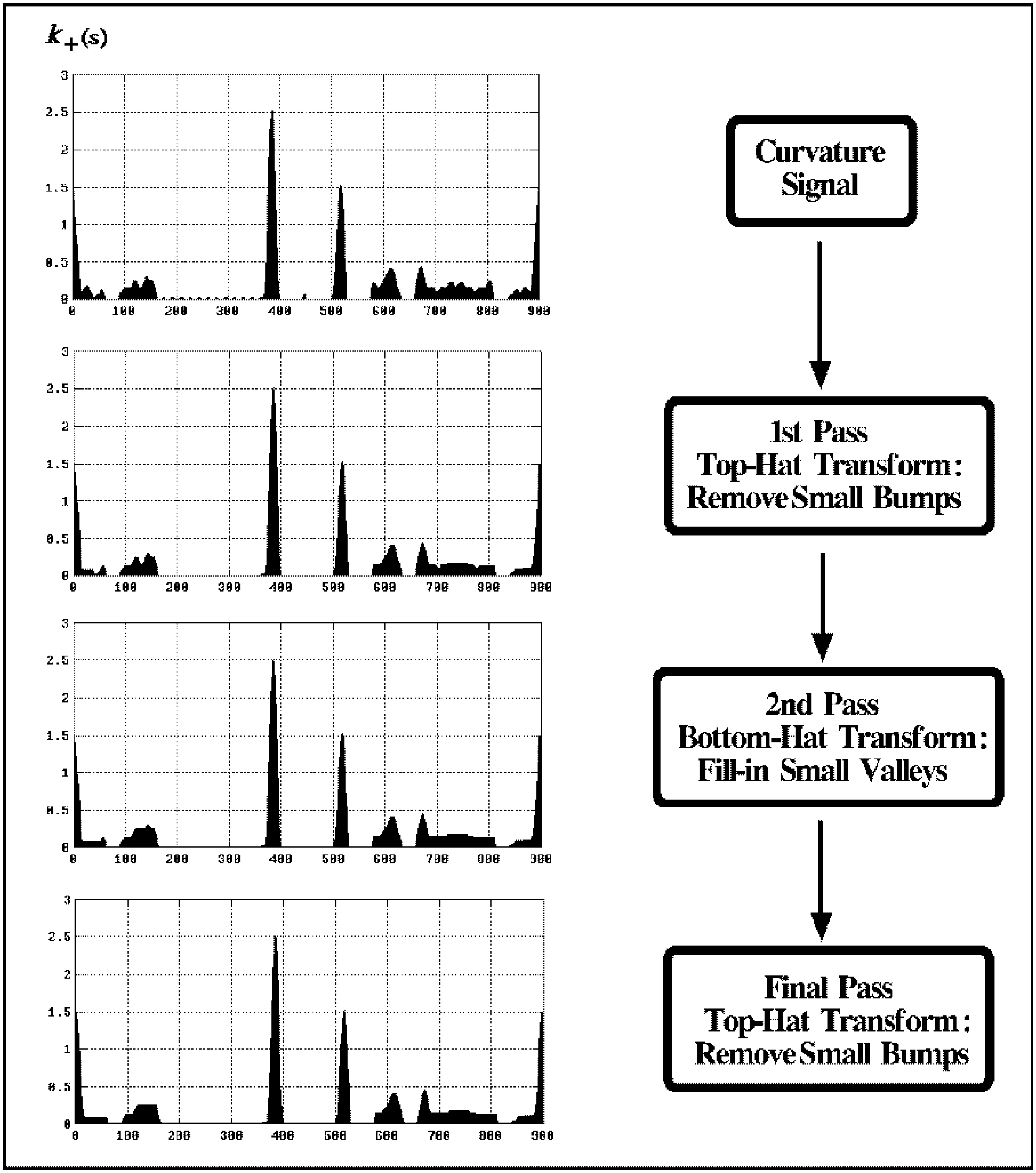
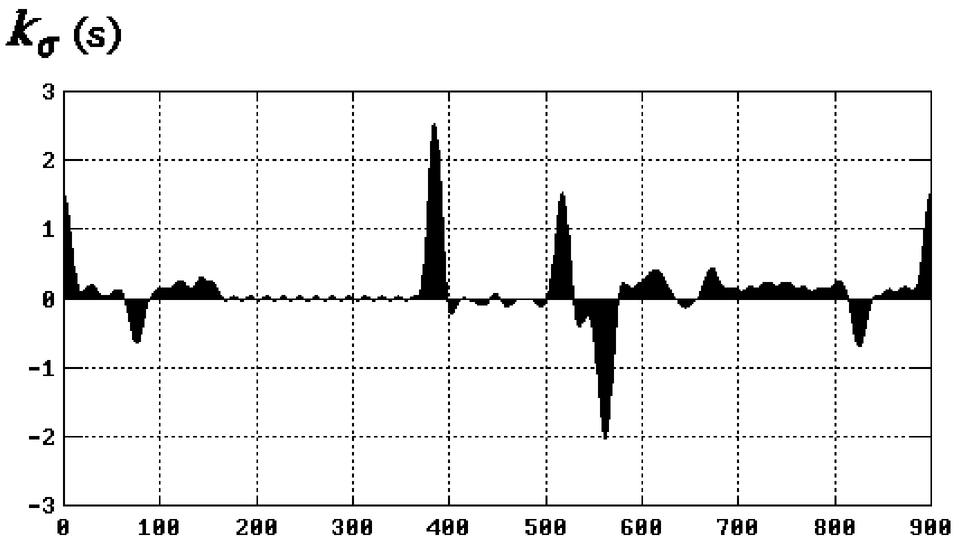
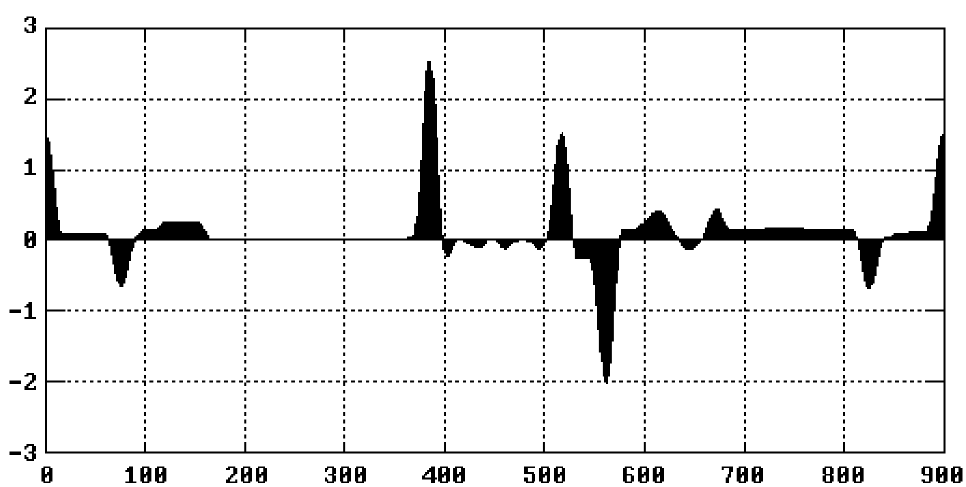


Figure 12: Removing noise from $k_+(s)$. At each one of the three passes, the MCS is used to extract small bumps and depressions.



(a) Original signal.



(b) Filtered signal.

Figure 13: Example of removing noise from $k_\sigma(s)$ using the hat transforms and the MCS.

on the other hand, can be classified as corresponding to constant curvature regions because their initial larger extent δ_s maps to a constant curvature arc. Once these peaks are extracted, they can be subtracted from the curvature signal. The resultant signal can then be further processed to isolate other constant curvature arcs. An example of this segmentation process applied to both positive and negative curvature signals is shown in Figures 14 and 15.

Once the main features of the curvature signal have been segmented, that is when both positive and negative peaks and constant curvature arcs have been extracted, the contour of the object itself can be segmented. Peaks of curvature map to knot points of the object contour, while constant curvature arcs map to their respective contour arcs. This is shown in Figure 16. All knot points are extracted and precisely localized.[13] Constant curvature arcs are also well identified. Gaps between identified arcs and knot points which are caused by filtering can be filled-in by extrapolation, while unidentified arcs can be approximated by fitting splines or circular arcs between the bounding knots [20]. The average curvature of these unidentified arcs, obtained by integrating the curvature signal between the knots, can be used to impose a constraint on the fitting curve segment [20].

3.4 Multidimensional Morphological Curvature Scale-Space

Aside from the history of peaks of $k_\sigma(s)$ in the MCS defined in the previous section, we can define other measures to be tracked in scale-space. Specifically, we propose using the morphological measures defined in § 3.3. For example, the shape factor r could be useful for tracking in scale-space a curvature peak that has widened at its base. We might also observe the rate of growth of k_{max} and k_{avg}. We are currently conducting experiments with the tracking of these measures in MCS. A multidimensional approach is then produced by combining these various morphological measures. We believe that this multidimensional MCS is a more powerful tool for the interpretation of curvature than previous scale-space approaches. There are four reasons: first, MCS is uniform and unambiguous; second, MCS satisfies the three criteria of causality, immediate localization and piecewise smoothing; third, morphological measures can be combined in MCS to strengthen the analysis and interpretation of curvature primitives; and fourth, MCS is strictly a local representation. Because of this last point we are able to extract local features without modifying the global shape of the curvature function. Only those regions having details smaller than or equal in size to the width of the structural element are modified.

Another important aspect of curvature morphology that compares favorably with other analysis approaches is its low computational complexity (see Appendix B). The computations are simple and hierarchical, and lend themselves to applications where fast computations are imperative. This is the case when sequences of images are being analyzed [28, 19].

[13] Note that no knot point is identified between arcs 2 and 3 because there is no discontinuity in orientation between these two arcs. Therefore, there is no peak in curvature where these two arcs join. In such cases a *smooth join* can be identified [20].

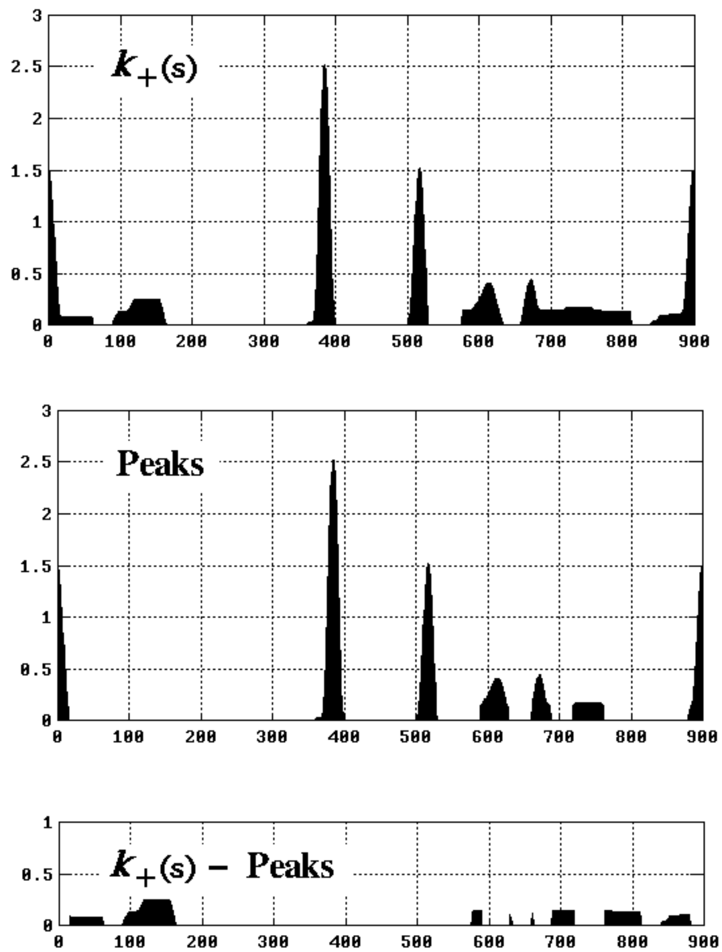


Figure 14: Segmentation of the positive curvature signal.

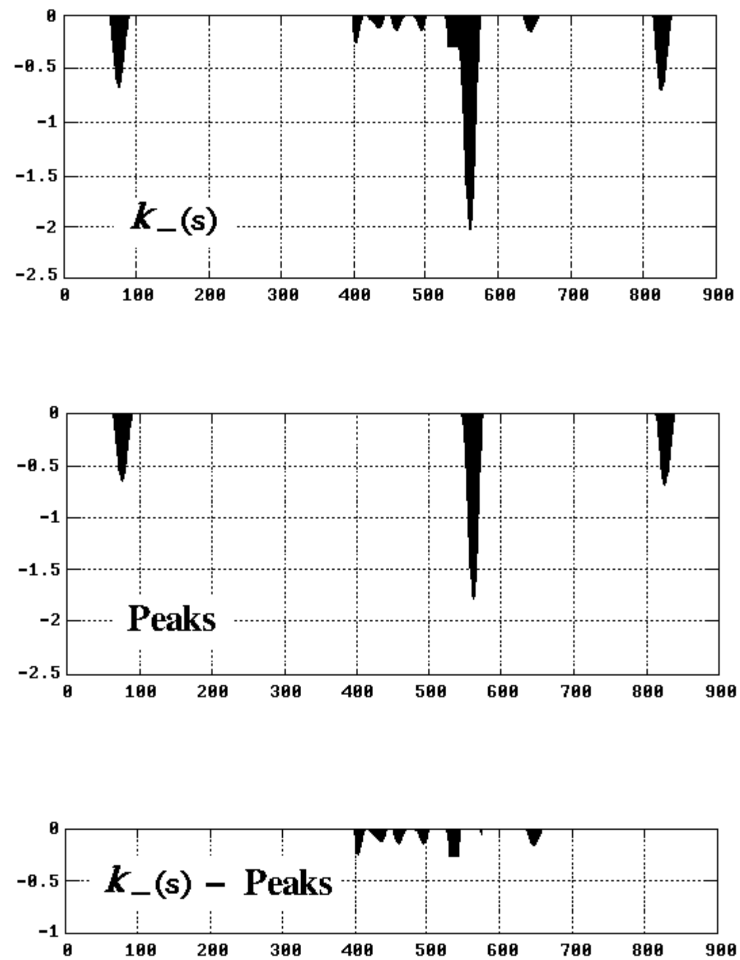
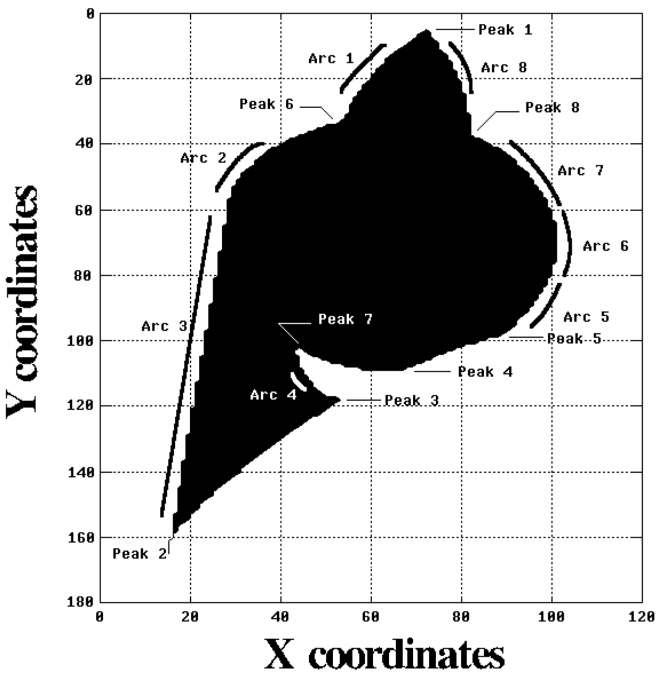


Figure 15: Segmentation of the negative curvature signal.



Positive Peaks

Peak#	Smax	X	Y
1	899	72	5
2	384	15	165
3	516	53	118
4	613	66	109
5	672	90	97

Negative Peaks

Peak#	Smax	X	Y
6	74	53	33
7	561	44	102
8	824	82	37

Constant Curvature Arcs

Arc#	S1	X	Y	S2	X	Y	Kavg
1	16	67	10	62	56	27	0.0786
2	112	38	41	158	28	58	0.2554
3	165	28	62	358	17	152	0.0000
4	575	48	105	589	54	108	-0.1319
5	687	93	93	717	99	81	0.1379
6	718	100	80	760	100	60	0.1764
7	763	99	59	813	87	41	0.1345
8	851	81	25	890	75	9	0.0812

Figure 16: Contour segmentation based on the previously extracted curvature features.

4 Conclusions

In this paper we have proposed efficient ways of representing curvature and its features for object contours found in noisy images. We have discussed how the curvature function should be retrieved from the discrete trace of a contour using a modified chain code representation. Special care has been taken with quantization errors, "protrusion-depression definition" problems, and implementation issues. We have also presented a new method for extracting features from the curvature function employing morphological operations.

Unlike linear transformations of functions such as Gaussian filtering, morphological operations are characterized by their local effect. They remove information of increasing extent as the size of the structural element increases without blurring the remaining important features. Signal processing through iterative morphological transformations can therefore be thought of as a process of selective information removal, where irrelevant details are subtracted from the signal, enhancing the contrast of essential features. We have proposed such an iterative scheme by defining a new scale-space representation referred to as the Morphological Curvature Scale-Space. This representation for curvature satisfies the essential criteria of causality, immediate localization and piecewise smoothing.

A Morphological Operations for Functions

Let f be a function representing a signal, such as $k(s)$, the curvature along a contour. This function is defined in Euclidean 2-D space as a set of points $(s, f(s))$. Let B be a (2-D) structural element indexed by a single parameter i. The superscript c in f^c stands for the complement of f such that $f^c + f$ = constant; $\overline{B}$ stands for the reflection of B, that is $\overline{B} = \{i : -i \in B\}$.

A.1 The Four Principal Operations

A.1.1 Erosion

An erosion is computed by taking the minimum of a set of differences. Its form is similar to correlation, with the summation of correlation replaced by the minimum operation and the product replaced by a subtraction operation. The erosion of f by B is defined as follows:

$$E_{fB} \equiv f \ominus B = \min_i \left[f(s-i) - \overline{B}(i) \right] .$$

A.1.2 Dilation

Dilation can be performed by taking the maximum of a set of sums. Its complexity is the same as erosion and is related to convolution, where instead of doing summation of products, a maximum of sums is computed. Using the same notation as for erosion, the dilation of f by B is defined as:

$$D_{fB} \equiv f \oplus B = \max_i \left[f(s-i) + B(i) \right] .$$

Significance of the Erosion and Dilation Operations Erosion and dilation are dual operations but are not, in general, the inverse operation of each other. This is expressed below:

$$\begin{aligned} f \oplus B \ominus B &= f_1 \neq f \\ f \ominus B \oplus B &= f_2 \neq f \end{aligned}$$

Furthermore, in general

$$f_1 \neq f_2 .$$

This is true whenever the function f is not too smooth or regular. A dilation (erosion) will remove from the top (underneath) of f all those details that are smaller than the structural element B which is of fixed size. The result is a new function f' which is more regular or smoother than the original function f .

When the protrusions of f are not covered by the structural element B, then the combination of an erosion and a dilation becomes be a reversible operation (*i.e.*, $f_1 = f_2 = f$).

Therefore, a combination of an erosion and a dilation can give some information about the regularity of a function. Combinations of this type give rise to two new operations, which are defined next.

A.1.3 Opening

Opening is defined as the dilation of an eroded function. It is given by the following relation:

$$f_B \equiv (f \circ B)_{(i)} \equiv (f \ominus B) \oplus B = \sup \left\{ \inf \left[f(z) : z \in B_y \right] : y \in \overline{B_i} \right\} . \tag{7}$$

At the point i, $(f \circ B)_{(i)}$ has the highest value of the infinima of f taken over all the B's containing i. This means that the opening of f is a new function defined by the *highest* points reached by any part of the 2-D (reflected) structural element as it slides *under* the whole extent of f [36].

A.1.4 Closing

Closing is defined by the following relation:

$$f^B \equiv (f \bullet B)_{(i)} \equiv (f \oplus B) \ominus B = \inf \left\{ \sup \left[f(z) : z \in B_y \right] : y \in B_i \right\} . \tag{8}$$

The closing of a function can be interpreted as a new function defined by the *lowest* points reached by any part of the 2-D structural element as it slides *over* f [36].

A.2 Properties of the Four Operations

The principal properties of the operations discussed above are briefly discussed in this section. These properties are important in order to understand the effects of erosion, dilation, opening and closing on functions.

A.2.1 Increasing

Let f be a function smaller than another function F, that is $f(s) \leq F(s)$, for all s. Then:

$$\begin{aligned} f \ominus B &\leq F \ominus B \\ f \oplus B &\leq F \oplus B \\ f \circ B &\leq F \circ B \\ f \bullet B &\leq F \bullet B \end{aligned}$$

The size of a function implies the size of the eroded or dilated function for a given structural element B. Erosion and dilation are said to be increasing operations. Since erosion and dilation are increasing operations, opening and closing are also increasing.

A.2.2 Expansivity

$$\begin{aligned} f \ominus B &\leq f \\ f \oplus B &\geq f \\ f \circ B &\leq f \\ f \bullet B &\geq f \end{aligned}$$

Erosion is antiexpansive, while dilation is expansive. The terms "erosion" and "dilation" have their origins in this property of expansivity. Opening is antiexpansive, as is its first constituent operation, erosion. Closing is expansive, as is dilation.

A.2.3 Duality

$$\begin{aligned} f \oplus B &= \left(f^c \ominus \overline{B}\right)^c \\ f \ominus B &= \left(f^c \oplus \overline{B}\right)^c \\ f \circ B &= \left(f^c \bullet \overline{B}\right)^c \\ f \bullet B &= \left(f^c \circ \overline{B}\right)^c \end{aligned}$$

Dilation is the dual of erosion. Thus dilation (erosion) is the erosion (dilation) of the complemented function. Opening and closing are also dual operations. Closing of f corresponds to the opening of the complemented function.

A.2.4 Chain Rule

$$\begin{aligned} (f \oplus B_1) \oplus B_2 &= f \oplus (B_1 \oplus B_2) \\ (f \ominus B_1) \ominus B_2 &= f \ominus (B_1 \oplus B_2) \end{aligned}$$

This implies that the erosion or dilation of a function f by a wide or complex structural element $B \equiv B_1 \oplus B_2$ can be performed using the basic components of B, namely B_1 and B_2. An image processor can then be built using only basic components to perform any kind of dilation or erosion.

A.2.5 Idempotency

$$\begin{aligned} (f \circ B) \circ B &= f \circ B \\ (f \bullet B) \bullet B &= f \bullet B \end{aligned}$$

Opening and closing operations are performed only once for a specific structural element. Their is no equivalence to the chain rules of erosion and dilation. As discussed previously, opening and closing are two ways to compute a "smooth" approximation of a function f by removing details of a given size. Applying these operations again with the same structural element B does not further modify the filtered function.

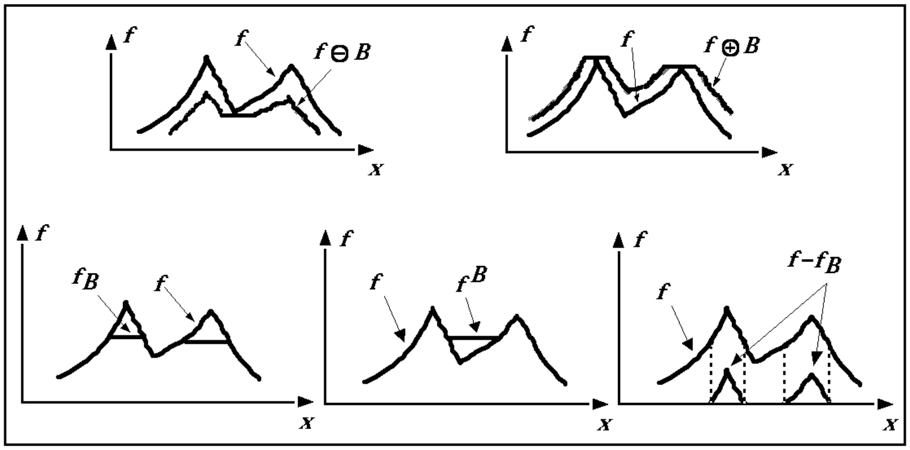


Figure 17: Examples of the four morphological operations and of the Top-Hat transform (adapted from [36]).

A.3 Significance of the Morphological Operations

Unlike linear transformations of functions, morphological operations are characterized by being non-invertible. They remove information of greater and greater extent as the size of the structural element increases.

Signal processing through iterative morphological transformations can therefore be conceived as a process of selective information removal where irrelevant details are irrecoverably destroyed, thereby enhancing the contrast of essential function features [37].

A.4 Application to Curvature: Flat Structural Element

For the morphological analysis of a signal such as curvature where we are interested in the extraction of peaks and constant regions, flat structural elements (1-D) are used. Eroding a function by a segment of size $B = R$ is equivalent to replacing the function values at every point by the minimum of all the points in a neighborhood of radius R. Likewise, dilating a function by a segment of size $B = R$ is equivalent to a maximum transformation over a neighborhood of radius R. Openings and closings by flat structural elements maintain the vertical boundaries of the function they transform.

Hat-transforms, so named because they can be visualized as a covering of peaks with a hat of fixed size, can be used to extract sharp peaks and ridges. The residual $f - f_B$ (where f_B is the opened version of f as defined in equation 7) is known as the Top-Hat transform and presents the possibility of extracting peaks. Its complement, the residual $f - f^B$ (where f^B is the closed version of f as defined in equation 8), is known as the Bottom-Hat transform and provides a way to extract valleys. Examples of the four morphological operations and of the residual $f - f_B$ with a flat structural element, are given in Figure 17.

Morphological analysis of the residuals $f - f_B$ and $f - f^B$, and of the filtered signals f_B and f^B, can then be performed to extract and classify peaks, valleys and the resultant flattened regions.

B Issues of Computational Complexity

Morphological operations for a function such as $k(s)$, using a flat structural element B are simple and easy to implement as $\min$-$\max$ operations. Erosion is equivalent to taking the minimum of $k(s)$ over the neighborhood defined by the width of B, while dilation is equivalent to taking the maximum of $k(s)$ over the same neighborhood (see previous appendix).

Both erosion and dilation can be implemented as iterative processes by using the *chain rule* property (see § A.2). This property implies that an erosion or dilation with a relatively wide structural element (*e.g.*, 5 pixels wide) can be performed sequentially with smaller structural elements (*e.g.*, two structural elements 3 pixels wide), giving a more efficient way to perform $\min$-$\max$ operations. However, the chain rule property does not apply directly to the opening and closing operations. These two are said to be *idempotent* (see § A.2). That is, their application to a given function with the same structural element does not further modify the filtered function. Thus we can only take advantage of the chain rule property for the first operation of an opening or a closing, that is, an erosion or a dilation, respectively. Using the latter for the first constitutive operation of closing and opening operation a hierarchical-like scheme can be implemented. For example, consider the hierarchical implementation of an opening operation. Let the basis of the hierarchy be the positive curvature function $k_+(s)$. Subsequent levels are build-up of eroded and opened versions of $k_+(s)$: $k_{+erod}(s,l)$ and $k_{+open}(s,l)$, where l corresponds to the hierarchical level and $k_{+open}(s,l)$ is obtained by dilating $k_{+erod}(s,l)$. Therefore, an eroded version of $k_+(s)$ at a particular level in the hierarchy, that is, $k_{+erod}(s,l)$, can be obtained efficiently (*i.e.*, with a small structural element) by eroding an eroded version of $k_+(s)$ at a precedent level, that is, by the erosion of $k_{+erod}(s,l-1)$. Such a hierarchical scheme gives an efficient way of building the MCS, where each level in the hierarchy corresponds to an increasing δ_s.

Morphological measures are also simple to implement. Furthermore, due to the uniformity of MCS features, morphological measures are more rapidly computed for existing peaks, that is, for peaks that were detected in a precedent level in the hierarchy or at a smaller scale δ_s, since their localization is already known. Finally, this property of uniformity permits simple and easy interpretation, a major advantage in terms of computations over traditional scale-space approaches.

In summary, three aspects of curvature morphology lead to a low computational complexity figure. Firstly, only simple computations for both morphological operations and measures are necessary. Secondly a hierarchical implementation using the chain rule property is easily obtained. Finally, the uniformity of the MCS and its simplicity of interpretation yields low computational complexity for the *Curvature Morphology* representation.

References

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:2–14, Jan. 1986.

[2] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, U.S.A., 1987.

[4] A. Blake, A. Zisserman, and A. V. Papoulias. Weak continuity constraints generate uniform scale-space descriptions of plane curves. In *Advances in Artificial Intelligence*, pages 587–597. Elsevier Science, 1987.

[5] C. Blakemore and R. Over. Curvature detectors in human vision. *Perception*, 3:3–7, 1974.

[6] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3:36–61, 1984.

[7] P. J. Burt. Fast filter transforms for image processing. *CGIP*, 16:20–51, 1981.

[8] L. P. Cordella and G. Dettori. An $O(N)$ algorithm for polygonal approximation. *Pattern Recognition Letters*, 5(3):93–97, March 1985.

[9] J. L. Crowley and R. M. Stern. Fast computation of the difference of low-pass transform. *IEEE Transactions on PAMI*, 6(2):212–222, March 1984.

[10] A. R. Dill, M. D. Levine, and P. B. Noble. Multiple resolution skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):495–504, July 1987.

[11] Manfredo do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[12] A. Dobbins, S. W. Zucker, and M. Cynader. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329:438–441, Oct. 1987.

[13] M. M. Fleck. Local rotational symmetries. In *Proceedings of the 1986 Conference on Computer Vision and Pattern Recognition*, pages 332–337, Miami, Florida, June 1986. IEEE Computer Society.

[14] H. Freeman. On the classification of line drawing data. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 408–412. M.I.T. Press, Cambridge, MA, 1967.

[15] H. Freeman and A. Saghri. Generalized chain codes for planar curves. In *Proceedings of the 4^{th} Joint International Conference on Pattern Recognition*, pages 701–703, Kyoto, Japan, Nov. 1978. IEEE Computer Society.

[16] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[17] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[18] M. D. Levine. *Vision in Man and Machine*. Computer Engineering Series. McGraw-Hill, NY, 1985.

[19] F. Leymarie and M. D. Levine. New method for shape description based on an active contour model. In W. A. Pearlman, editor, *Proc. of SPIE Visual Communications and Image Processing*, volume 1199, part 1, pages 390–401, Philadelphia, PA, U.S.A., November 1989.

[20] F. Leymarie and M. D. Levine. Shape features using curvature morphology. In D. P. Casasent, editor, *Proc. of SPIE Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, volume 1192, part 2, pages 536–547, Philadelphia, PA, U.S.A., November 1989.

[21] F. Leymarie and M. D. Levine. Snakes and skeletons. McRCIM Technical Report CIM–89–3, McGill Research Center for Intelligent Machines, McGill University, Montreal, Canada, January 1989. 29 pages.

[22] N. K. Link and S. W. Zucker. Corner detection in curvilinear dot grouping. *Biological Cybernetics*, 59:246–256, 1988.

[23] A. Mackworth and F. Mokhtarian. The renormalized curvature scale space and the evolution properties of planar curves. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, pages 318–326, Ann Arbor, Michigan, June 1988. IEEE Computer Society.

[24] J. W. McKee and J. K. Aggarwal. Computer recognition of partial views of curved objects. *IEEE Transactions on Computers*, C-26:790–800, Aug. 1977.

[25] G. Medioni and Y. Yasumoto. Corner detection and curve representation using cubic B-splines. *CVGIP*, 39(3):267–278, 1987.

[26] P. Meer, E. S. Baugher, and A. Rosenfeld. Extraction of trend lines and extrema from multiscale curves. *Pattern Recognition*, 21(3):217–226, 1988.

[27] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:34–43, Jan. 1986.

[28] P. B. Noble and M. D. Levine. *Computer-Assisted Analyses of Cell Locomotion and Chemotaxis*. CRC Press, Boca Raton, Florida, 1986.

[29] L. O'Gorman. An analysis of feature detectability from curvature estimation. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, pages 235–240, Ann Arbor, Michigan, June 1988. IEEE Computer Society.

[30] A. V. Oppenheim and R. W. Schafer. *Discrete–Time Signal Processing*. Prentice-Hall.

[31] P. Parent and S. W. Zucker. Trace inference, curvature consistency and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:823–839, Aug. 1989.

[32] T. Pavlidis. Curve fitting as a pattern recognition problem. In *Proceedings of the* 6^{th} *ICPR*, pages 853–857, Munich, Germany, 1982. IEEE CS Press.

[33] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, C-23(8):860–870, August 1974.

[34] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *Proceedings of the 1987 IEEE Workshop on Computer Vision*, pages 16–22, Miami Beach, Florida, Nov. 1987. IEEE Computer Society.

[35] A. Rosenfeld and Weszka. An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, C-24:940–941, 1975.

[36] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[37] S. R. Sternberg. Grayscale morphology. *Computer Vision, Graphics and Image Processing*, 35:333–355, 1986.

[38] K. Wall and P.-E. Danielsson. A fast sequential method for polygonal approximation of digitized curves. *CVGIP*, 28:220–227, 1984.

[39] W. M. WellsIII. Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Transaction on PAMI*, 8(2):234–239, March 1986.

[40] A. P. Witkin. Scale-space filtering. In *Proceedings of the* 8^{th} *International Joint Conference on Artificial Intelligence*, pages 1019–1022, Kalsruhe, West Germany, Aug. 1983. AAAI.