Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Andrei C. Jalba · Michael H.F. Wilkinson · Jos B.T.M. Roerdink · Micha M. Bayer · Stephen Juggins

Automatic Diatom Identification using Contour Analysis by Morphological Curvature Scale Spaces

Received: date / Revised version: date

Abstract A method for automatic identification of diatoms (single-celled algae with silica shells) based on extraction of features on the contour of the cells by multi-scale mathematical morphology is presented. After extracting the contour of the cell, it is smoothed adaptively, encoded using Freeman chain code, and converted into a curvature representation which is invariant under translation and scale change. A curvature scale space is built from these data, and the most important features are extracted from it by unsupervised cluster analysis. The resulting pattern vectors, which are also rotation-invariant, provide the input for automatic identification of diatoms by decision trees and k-nearest neighbour classifiers. The method is tested on two large sets of diatom images. The techniques used are applicable to other shapes besides diatoms.

Keywords Diatom identification, mathematical morphology, contour analysis, curvature scale spaces, multi-scale analysis, decision trees.

1 Introduction

In both human and computer vision systems, curvature extrema of the contour of an object are thought to contain important information about the shape [26]. In this paper we present a technique to extract this information by multi-scale mathematical morphology, and use this for automatic identification of diatoms by decision trees and k-nearest neighbour classifiers.

The shapes considered in this paper are outline images of diatoms, though the methodology might be applied to many other shapes. Diatoms are microscopic, single-celled algae, which build highly ornate silica shells or frustules. Some examples are shown in Fig. 1. Apart from being highly ornamental, the shapes and ornamentations of these shells allow phycologists to classify and identify these organisms. Furthermore, these shells fossilize well, allowing the study of fossil as well as living diatoms. Diatoms occur in almost any aquatic habitat, ranging from planktonic diatoms in lakes and the oceans, to benthic species which are either firmly attached to some surface on the sea or lake floor, or occur more freely between the particles of a silty or sandy sediment. The sensitivity of certain diatom species to environmental parameters means that they can be used to monitor changes in the environment, or have forensic applications. All these applications require counting and identification of different species of diatoms present in the sample of interest. This usually involves many hundreds of cells per slide. Apart from being very tedious work, accurate identification requires a high degree of skill and experience, which is not always available in routine laboratory settings. Therefore automation of the identification of diatoms by image analysis is highly desirable. The Automatic Diatom Identification and Classification (ADIAC) project [5], of which this research is a part, aims to automate the process of diatom identification by digital image analysis. For further details on diatom morphology and applications, see [36, 40].

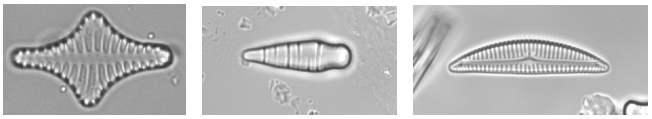

Fig. 1 Some examples of diatom shells.

The starting point for automated analysis is formed by 8-bit greyscale images acquired using brightfield light microscopy and a 1018×1008 pixel CCD camera. Most images were captured with a 100x objective, with most taxa lying between 5 and 200 microns in length, giving images of 600×400 pixels on average. The images vary in quality, contours being quite noisy. Any automatic identification method must be able to deal with such noise (a point which

Andrei C. Jalba · Michael H.F. Wilkinson · Jos B.T.M. Roerdink
Institute for Mathematics and Computing Science
University of Groningen
P.O. Box 800, 9700 AV Groningen, The Netherlands

Micha M. Bayer
Royal Botanic Garden Edinburgh
Scotland, UK

Stephen Juggins
University of Newcastle upon Tyne, UK

is extensively discussed in section 2.2). The contours which we use here were extracted during the ADIAC project [5] using the automatic method by Fischer et al. [12]. The overall procedure for the segmentation of diatom images is as follows [12]. In a pre-processing step, multiple objects from an image are detected by background suppression. The remaining regions are analyzed in detail (using threshold selection, local texture analysis and graph-based merging) to find the exact location of diatom contours. After pre-processing, the locations of the boundaries between diatoms and background is determined using two difference contour extraction methods. The first method automatically selects global thresholds in order to label the dark diatom border. The second method employs an edge detector in order to extract all edge information. One of these methods has to be chosen depending on the quality of the input image, such that binary images with closed contours are always obtained. In a final step, these closed contours are extracted using a conventional contour-following algorithm.

Several techniques for multi-scale shape analysis exist, such as size distributions, or granulometries, which are used to quantify the amount of detail in an image at different scales [3, 32]. A similar method, based on sequential alternating filters, has been proposed by Bangham and coworkers [1, 2]. Their method is used on 1-D signals, though they do discuss extensions to higher dimensions. Mokhtarian and Mackworth [30, 31] showed that curvature inflection points extracted using a Gaussian scale space can be used to recognize curved objects. One difficulty with this approach is that curves without inflection points fall into the same equivalence class. In this paper we develop a different multi-scale approach to the analysis of 1-D signals, based on work by Leymarie and Levine [25]. They developed a morphological curvature scale space for shape analysis, based on sequences of morphological top-hat or bottom-hat filters with increasing size of the structuring element used. A problem not addressed by Leymarie and Levine is that of extracting the most important features from the scale space. We will present some modifications of their technique, and include a method by which the features in the scale space may be clustered in an unsupervised way. The aim is to obtain a small set of rotation, translation and scale-invariant shape parameters, which contain as much information about the shapes of interest as possible.

The remainder of this paper is organized as follows. Section 2 describes the construction of the curvature scale spaces by means of mathematical morphology. It gives some modifications of the original method proposed by Leymarie and Levine [25]. We focus also on the curvature measure extraction, which poses some noise related problems, and on the adaptive smoothing method employed to deal with these problems. Section 3 describes the clustering method used for the extraction of the most important features from the curvature scale spaces, which is one of the main purposes of this work. Experimental results are reported in Section 4 with a discussion of important observations. Conclusions are drawn in Section 5.

2 Morphological curvature scale spaces

In this section we describe how the curvature is extracted from the contour of the diatoms, give the construction of the curvature scale spaces and show how feature vectors are obtained from the scale-space signals.

2.1 Curvature based shape recognition

A common approach to shape recognition is by focusing on the curvature of the contour [34]. Since the curvature of the contour defines a shape completely, it should in theory be possible to classify diatom shapes using the curvature. An obvious way is to use Fourier descriptors [16, Sect. 8.1.1] of the curvature function, but these have a number of drawbacks. The most notable one is that each Fourier descriptor contains information about all parts of the contour, i.e., localization of features is lost. Even global properties of the contour such as symmetry can be difficult to obtain from Fourier descriptors in the presence of noise. Methods based on mathematical morphology [1, 2, 25] do not suffer from the delocalization problem, and for this reason they will be explored in this paper.

2.2 Curvature measure extraction

Whatever the method used, all start out by computing a contour representation of the object. Various methods to compute the curvature along the contour of an object use Freeman's chain-code representation of the contour as a starting point. The Freeman chain code is a compact way to represent the contour of an object that uses for coding three bits per pixel. The chain code is a sequence of n links c_i, $i = 1, 2, \ldots, n$, where c_i is a vector connecting neighboring edge pixels. The directions of c_i are coded with integer values $k = 0, 1, \ldots, 7$ in a counterclockwise direction starting from the direction of the positive *x*-axis.

Differences between adjacent contour codes can in principle yield curvatures, but this direct approach results in a highly noisy curvature measure, due to the small number of possible directions of travel (four or eight). Therefore, in order to correct for the quantization problems inherent to the chain code representation we follow the approach by [25]. For obtaining reliable estimates of image measurements (e.g. perimeter, area, moments, curvature, etc.) based on contour information, one must take into account the noisy nature of binary contours due to many factors, from discrete sampling and quantization errors to segmentation errors. A frequently used method for dealing with these types of errors, before extracting some useful information, is to smooth the binary contours first. In general, the purpose of smoothing is twofold: noise is eliminated to facilitate further processing, and features irrelevant to a given problem are ruled out to reduce the complexity.

One common method of smoothing is to filter the contour using a Gaussian kernel $G_\sigma(x)$, with width σ, defined as

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-x^2}{2\sigma^2}. \quad (1)$$

If we denote by $\theta(x)$ the chain code, then the curvature $k(x)$ is given by

$$k(x) = (\theta * G_\sigma)'(x) = (\theta * G'_\sigma)(x). \quad (2)$$

Here $*$ denotes convolution, and the prime indicates differentiation. This simple method of smoothing did not prove to be suitable when applied to diatom contours [44], because in homogeneous regions it undersmoothed, while in strongly curved regions oversmoothing occurred. Since our purpose is to extract the most important curvature features of the contour, and the curvature signal is very noisy, we need to apply *adaptive* smoothing according to a suitable criterion in order to perform a tradeoff between the loss of information due to smoothing and the noise reduction. We choose to obtain as much as possible information in the strongly curved regions, where the mean curvature in a given window has a large value, and to increase the degree of smoothing as the mean curvature decreases (i.e., in weakly curved regions). Therefore, we need to build a function for varying the width of the derivative Gaussian kernel in every point along the contour, which decreases when the mean curvature increases and vice-versa. A simple example of such a function is given by:

$$\sigma = \sigma_{min} + \frac{c_1 \cdot \sigma_p}{1 + c_2 \cdot \mu} \quad (3)$$

where σ_{min} is the minimum width of the smoothing kernel and μ is the mean curvature computed in a fixed window of width W obtained after a 'pilot' smoothing of the contour with a fixed width of the kernel σ_p, as explained below. The two constants c_1 and c_2 are obtained as follows:

1. We compute c_1 by imposing the condition that the width of the kernel (σ_{max}) is maximal when the mean pilot curvature in the window is 0:

$$c_1 = \frac{\sigma_{max} - \sigma_{min}}{\sigma_p} \quad (4)$$

2. The constant c_2 is obtained by imposing the condition that the width of the kernel equals twice the minimal value σ_{min} when the mean curvature in the window is μ_0. The condition is:

$$\sigma_{min} = \frac{c_1 \cdot \sigma_p}{1 + \mu_0 \cdot c_2} \quad (5)$$

where c_1 is given by (4). Then the constant c_2 is:

$$c_2 = \frac{\sigma_{max} - 2 \cdot \sigma_{min}}{\mu_0 \cdot \sigma_{min}}; \quad (6)$$

The expression of the function is then:

$$\sigma = \sigma_{min} + \frac{\sigma_{max} - \sigma_{min}}{1 + \frac{\sigma_{max} - 2 \cdot \sigma_{min}}{\mu_0 \cdot \sigma_{min}} \cdot \mu} \quad (7)$$

The function is plotted in Fig. 2 for three values of μ_0. In all experiments below, a value of $\mu_0 = 0.015$ is used. The values for the other three constants ($\sigma_{min} = 0.02$, $\sigma_{max} = 0.08$ and $\sigma_p = 0.035$) were obtained empirically in such a way that the best identification performance was obtained.

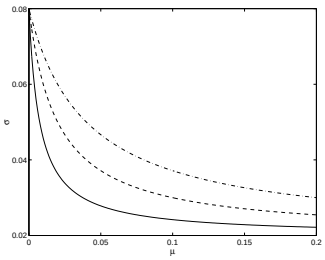

Fig. 2 Computation of the width of the derivative Gaussian kernel; *continuous*: $\mu_0 = 0.015$, *dashed*: $\mu_0 = 0.04$, *dashed-dotted*: $\mu_0 = 0.08$.

The steps of the adaptive smoothing method can be summarized as follows:

1. Smooth the contour with a derivative Gaussian kernel of fixed width $\sigma_p = 0.035$.
2. Compute the mean curvature in every point along the diatom contour, in a window of width W. The window width (W), also obtained empirically, is a fraction (5 %) of the length of the diatom contour.
3. Evaluate the width of the kernel using the function given in (7) in every point along the contour.
4. Smooth the contour a second time, but now adaptively, using the newly computed values for the width of the kernel.

In Fig. 3 we show some diatom outline images and their curvature functions after ordinary and adaptive Gaussian smoothing. First of all, it is clear that the different diatom types shown here each have highly distinctive curvature patterns. To demonstrate the advantage of adaptive Gaussian smoothing we show in Fig. 4 an enlargement of two regions in the curvature function of the left diatom image in Fig. 3, i.e., the intervals [0.2,0.3] and [0.3,0.6], respectively. The curvature function on the interval [0.2,0.3], which corresponds to the leftmost part of the diatom image, shows two distinct peaks in the adaptively smoothed signal, while ordinary smoothing has removed one of the peaks. In contrast, the curvature function on the interval [0.3,0.6], which corresponds to the slowly varying convex-concave-convex middle lower part of the diatom image, shows a nice smooth curve in the adaptively smoothed signal, while ordinary smoothing has failed to remove some of the small noise peaks.

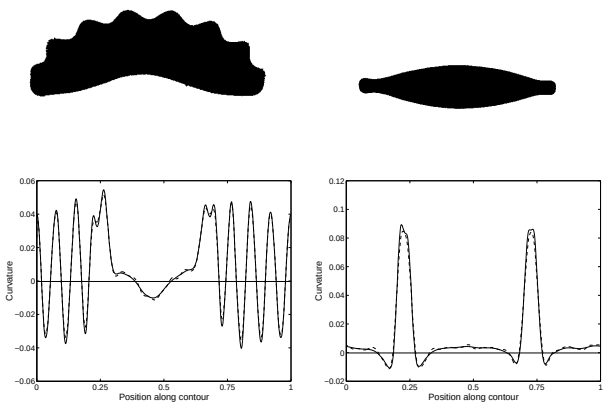

Fig. 3 Diatom images and their corresponding curvatures as function of position along the contour; *dashed* - Gaussian smoothing, *continuous* - Adaptive Gaussian smoothing.

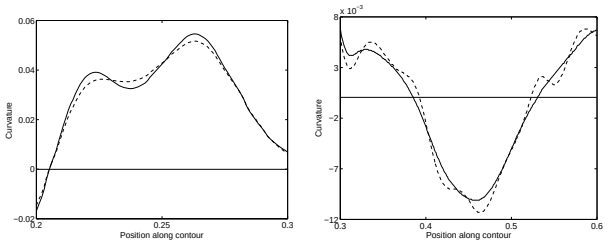

Fig. 4 Two enlargements of the curvature function of the left diatom image in Fig. 3 corresponding to the intervals [0.2,0.3] and [0.3,0.6], respectively; *dashed* - Gaussian smoothing, *continuous* - Adaptive Gaussian smoothing.

2.3 From curvature to scale-space features

The multi-scale representation is a very useful tool for handling image structures at different scales in a consistent manner [27], and there has been an increasing trend to use multiple scales in image analysis and computer vision. The basic idea is to embed the original signal into a stack of gradually smoothed signals, in which the fine scale details are successively suppressed.

Consider a signal $f : \mathbb{R}^n \to \mathbb{R}$ and a family of smoothing kernels $g_\sigma : \mathbb{R}^n \times \mathbb{R}$, where σ is the scale parameter. The signal smoothed at scale σ is $F : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$F(x, \sigma) = (f * g_\sigma)(x), \tag{8}$$

where $*$ denotes convolution. F is then a function in the $(n+1)$-dimensional space called *scale-space* and is known as the *scale-space image* of the signal [45].

Morphological operators [18, 38] can remove structure from a signal and therefore they were found suitable for scale-space smoothing. Jackway and Deriche [20] have demonstrated the application of reduced multiscale dilation-erosion fingerprints to surface matching. Chen and Yan [7] have used a scaled disk for the morphological opening of objects in binary images resulting in a theorem for zero-crossings of object boundary curvature. Park and Lee [33] have generalized the concept of zero-crossing for one-dimensional gray-scale signals.

In order to extract the convex and concave parts of the contour, corresponding to *peaks* and *valleys* of the curvature signal, we use a method adapted from [25]. This represents an alternative approach for the construction of curvature scale-space, in which the position and height of extrema in the signal are preserved with increasing scale until they vanish, rather than obtaining an 'evolution' of zero-crossings of the curvature.

The *hat-transforms* represent an important class of morphological transforms used for detail extraction from signals or images. Here we apply the hat-transforms to one-dimensional signals, i.e., the curvature functions of the diatom contours. Assume a signal f and a 1-D structuring element K. The opening of f by K is denoted by $f \circ K$. The residue of the opening compared to the original signal, i.e., $f - (f \circ K)$ represents the *top-hat* transformation. Those parts of the signal that do not fit into the structuring element K are removed by the opening. Thus, when the opened signal is subtracted from the original a signal is obtained which contains the desired detail (see Fig. 5). Its dual, the *bottom-hat* transform, is defined as the residue of a closing $f \bullet K$ compared to the original signal f. Therefore, one can use hat-transforms with increasing size of the structuring element to extract details of increasing size. By performing repeated hat-transforms with increasing size of the structuring element on the curvature signal, we can build the morphological curvature scale spaces. A curvature scale space consists of a number of levels $\ell = 1, 2, \ldots, L$, where each level ℓ corresponds to a top-hat transform with a size δ_ℓ, where δ_ℓ increases with ℓ. Let the smoothed curvature signal be stored in an array C, K_ℓ denote the structuring element used at level ℓ and T_ℓ denote the top-hat $C - (C \circ K_\ell)$. All nonzero elements of T_ℓ are parts of features at scales δ_ℓ or smaller. Starting at level 1, we apply top-hat transforms to extract peaks of maximum curvature. At each level ℓ, T_ℓ is compared to $T_{\ell-1}$. If a peak which is present at level $\ell - 1$ stops increasing at level ℓ, i.e. $T_\ell = T_{\ell-1}$, it is removed by subtraction from the original curvature signal C, and its extremal curvature value, mean curvature, extent and location are stored in a node of a doubly-linked list. This process ends when

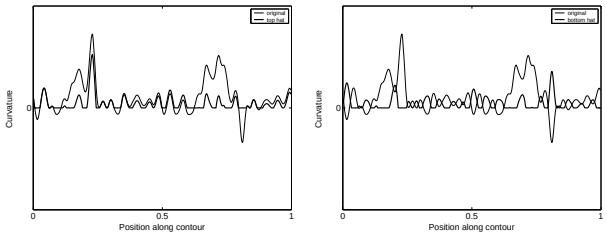

Fig. 5 Examples of top-hat and bottom-hat transform, showing the residues $f - (f \circ K)$ and $f - (f \bullet K)$, respectively, as a function of position along the contour. Here f is the original curvature function.

either all elements of array C are zero, or the largest scale L is reached. This yields the top scale space, in which every peak is precisely localized along the contour. A similar approach is used to obtain the bottom scale space (by performing bottom-hat transforms), in which all valleys are described. At the end of this we have obtained two curvature scale spaces, a top scale space of peaks and a bottom scale space of valleys.

Both scale spaces can contain spurious detail due to noise and therefore a filtering step is necessary to remove those features with curvature smaller than a given threshold (0.005 gave us the best results in the final clustering). Also, another approach we found useful for noise removal, is to increase the width of the structuring element at each level ℓ not by one, but by a fraction of the contour length. Note that because of the noise filtering it is not possible to reconstruct the original curvature signal from the scale-space data.

Another issue is that we need to deal with nested features (i.e. peaks of the signal). Nested features occur frequently in the data, e.g., when the ends of diatoms are more-or-less square. This is the case in the left-hand diatom in figure 3. Its ends are indicated by two broad peaks at positions 0.3 and 0.7 along the contour respectively, each with smaller peaks superimposed. The smaller peaks correspond to the sharper corners of either end of the diatom. Leymarie and Levine do not deal with this case, stating that only one curvature feature may be associated with a given section of the contour. The key difference between our approach and that of Leymarie and Levine is that we do allow multiple features to be associated with a given section of the contour. We do this simply by storing all features extracted by the algorithm in the linked list, sorted in ascending order by position. Using position and width information it is possible to determine whether peaks are nested, though we do not explicitly use this possibility in the further analysis yet.

A further important difference between our method and that of Leymarie and Levine is that, unlike their representation, we do not split the boundary in convex and concave parts, instead we run the hat-transforms on the entire signal.

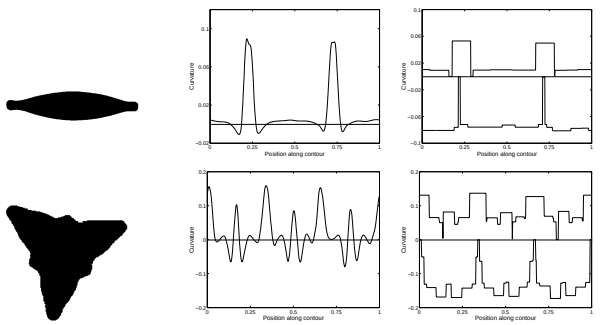

Fig. 6 Building curvature spaces. *Left*: binary images of diatoms; center: curvature plots; *right*: top and bottom scale spaces represented as curves, showing scale-space features as blocks of the correct width and average height.

The scale spaces can be visualized by plotting each feature as a box of either the maximum or the average height at the appropriate point in the curvature graph. If nested features are present, we can simply stack the features in the plot, as has been done in Fig. 6. The current method is sensitive to differences in the relative locations of the curvature features. This means that, for example, an elongated rectangle and a square are separable, because the widths of the major valleys in the curvature correspond to the distances between major peaks, so it is possible to discriminate between these. We gain scale invariance by computing the ratio between the extent of a peak (valley) and the contour length. Clearly, the extracted features are also translation-invariant.

3 Scale-space feature selection

This section is devoted to the extraction of a set of features by applying a clustering method to the curvature scale-space data. After some general remarks on data reduction, we describe the clustering method based on kernel density estimation, and construct two sets of feature vectors to be used for identification.

3.1 Reducing dimensionality of the scale-space data

Direct use of the scale-space data as pattern vectors for identification of shapes has several problems. The first is that the scale space may still contain spurious detail caused by noise. This increases the length of the pattern vector needlessly. Furthermore, the pattern vectors of different shapes would differ in length, which is a problem for many statistical methods. To standardize the pattern vector lengths, we either have to pad the shorter vectors with zeros, or we have to clip the longer vectors. In the latter approach, we use the N features with the largest maximum or average curvature. This automatically removes noise related features, which are expected to have low curvatures after the adaptive smoothing. However, we end up with the new problem of choosing N. If N is small, the pattern vector will not contain sufficient information, and it will be impossible to classify the shapes. If N is large, large numbers of images are needed to obtain good statistics. If we wish to compute Mahalanobis distances between clusters, each cluster must contain $N + 1$ data points *at least*, or the covariance matrices become singular.

One way to solve this problem would be to count the number of 'significant peaks' in the scale space. Again we run into problems, since we now have to decide which peaks are significant. The simplest approach is to use some threshold and count the number of peaks above the threshold. The problem reduces to finding an appropriate threshold. However, finding a single threshold suitable for all diatom shapes is by no means trivial. By selecting a threshold we are in effect creating a very coarse histogram of scale-space features with just two bins. An immediate extension of this idea is

to use more bins, to create a better description of the density distribution of the scale-space features. However, here we encounter another dimensionality problem: too few bins results in poor separation of features, too many means most bins are unpopulated.

A better approach would be to set the boundaries between classes of scale space features from the data themselves. This can be done by cluster analysis. However, no assumptions about the number of clusters or the shape of the distribution should be made *a priori*. Therefore an unsupervised clustering method should be used.

In the following subsections we will describe an unsupervised, distribution free method of cluster analysis, based on kernel density estimation. It should be stressed that any other unsupervised, distribution free method could be used as well.

3.2 Cluster analysis using the mean-shift algorithm

If we know that a d-dimensional data set is drawn from some known probability density distribution $p(\mathbf{x})$, an intuitive way of performing cluster analysis on such a data set is to use a gradient method and 'climb uphill' from each data point until a local maximum is reached. Each point can then be associated with a particular local maximum, and therefore a cluster. This intuitive idea can be put on a more formal basis, after we have described a method to estimate $p(\mathbf{x})$ from the data.

Kernel density estimates [39] approximate an unknown probability density distribution $p(\mathbf{x})$ from N data elements $\mathbf{x}_i$, by $\hat{p}(\mathbf{x})$, which is given by

$$\hat{p}(\mathbf{x}) = h^{-d} N^{-1} \sum_{i=1}^{N} K(\tfrac{\mathbf{x}_i - \mathbf{x}}{h}), \tag{9}$$

where $K(\mathbf{u})$ is called the kernel, d is the number of dimensions of $\mathbf{x}$, and h is the *window* or smoothing width. The kernel function is usually chosen as a non-negative, even function, with unity integral and unity variance [39]. The optimal, or automatic choice h_{opt} for the window width which minimizes the mean square integrated error (MSIE) can be estimated from

$$h_{opt} = \sqrt[d+4]{\frac{8(d+4)2^d \pi^{d/2}}{c_d N}}\sigma, \tag{10}$$

in which c_d is the volume of a d-dimensional unit sphere, and σ is the square root of the average of the variances of the data in each of the d dimensions. Robust estimators of σ are preferable, to reduce sensitivity to outliers. The optimal kernel function in terms of MSIE is the Epanechnikov kernel K_e [10], which is given by

$$K_e(\mathbf{u}) = \begin{cases} \frac{d+2}{2c_d}(1 - \|\mathbf{u}\|^2) & \text{if } \|\mathbf{u}\|^2 < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

This form of the Epanechnikov kernel does have unity integral, but its variance is $2d/(d+4)$. If we wish to compute the probability density estimate according to (9), we must divide the chosen window width by this factor first. However, here we are only interested in the limited support of K_e, and its parabolic form.

The gradient $\nabla\hat{p}(\mathbf{x})$ of the kernel density estimate $\hat{p}(\mathbf{x})$, which is needed for the 'hill climbing' algorithm, is given by

$$\nabla\hat{p}(\mathbf{x}) = h^{-d} N^{-1} \sum_{i=1}^{N} \nabla K(\tfrac{\mathbf{x}_i - \mathbf{x}}{h}). \tag{12}$$

Using K_e we have

$$\nabla\hat{p}(\mathbf{x}) = h^{-d} N^{-1} \sum_{\|\mathbf{x}_i - \mathbf{x}\| < h} \mathbf{x}_i - \mathbf{x}. \tag{13}$$

Therefore, the direction of ascent for any given location $\mathbf{x}$ is given as the mean of the vectors starting at $\mathbf{x}$ and ending at all *data points* $\mathbf{x}_i$ which lie within a radius of h of $\mathbf{x}$.

Based on this reasoning, Fukunaga and Hostetler [14] derived the following clustering algorithm (known in the computer vision community as the *mean-shift* algorithm [9]):

1. For each data item $\mathbf{x}_i$ compute the center of gravity $\bar{\mathbf{x}}_i$ of all data items which lie within h of $\mathbf{x}_i$ (including $\mathbf{x}_i$ itself!);
2. If, for any i, $\bar{\mathbf{x}}_i \neq \mathbf{x}_i$, replace all $\mathbf{x}_i$ by $\bar{\mathbf{x}}_i$, and repeat the first step;
3. Else end procedure.

Note that this is different from the k-means algorithm, since we only ever use those neighbours within radius h. Therefore, the number of neighbours used to assign a new value to each $\mathbf{x}_i$ may vary per iteration.

The method can be made unsupervised by using the automatic choice of smoothing width h_{opt} as in (10). In the experiments, we have simply used the standard method to estimate σ, i.e.

$$\sigma = \sqrt{\sum_{j=1}^{d} \frac{\langle x_j^2 \rangle - \langle x_j \rangle^2}{d}}, \tag{14}$$

where $\langle x \rangle$ stands for the expected value of x, and x_j is the jth component of vector $\mathbf{x}$. In the future we may implement more advanced methods.

As a final step in the curvature feature extraction, we construct two types of feature vectors:

Type 1 : for both top and bottom curvature scale spaces, select the first two clusters containing the scale-space features with the largest absolute curvatures, and for each cluster compute the number of peaks, mean curvature and variance.

Type 2 : for both top and bottom curvature scale spaces, select the first two clusters containing the scale-space features with the largest absolute curvatures, and for each cluster compute the mean curvature, and the extent and variance of the points with the highest curvature.

In addition, since curvature is a local attribute, two additional *curvature-related*, global shape descriptors are also computed. The first global curvature descriptor is the bending energy, defined as the sum of the squared curvatures along the contour. The second one is defined as the number of scale space entries from both top and bottom curvature scale spaces. Note that global shape descriptors were also used in [29,30] to supplement the Curvature Scale Space descriptors. Hence, in both cases the size of the feature vector is 14 ($2 \cdot 2 \cdot 3 + 2$). Since we no longer concern ourselves with the positions of points on the contour, the curvature measure is automatically rotation invariant. At the end of this a small set of rotation, translation and scale-invariant shape descriptors is obtained.

4 Experimental results

We applied our method to two sets of diatom contour files. The first set consists of 120 files, representing 6 demes (effectively separate species) of the *Sellaphora pupula* species complex. The second set, referred to as the *Mixed* data set, consists of 37 different taxa, comprising a total of 781 files. In both cases, each class (deme or taxon, respectively) has at least 20 representatives. Each file in the data sets represents a diatom outline for which we know the class (in the pattern recognition sense) that it belongs to. For the *Sellaphora* data set, classes are demes; for the *Mixed* data set, classes are species and varieties.

In our identification experiments we have used the C4.5 algorithm [35] for constructing decision trees, with bagging [4] as a method of improving the accuracy of the classifier. The performance was evaluated using the holdout [24] method. In addition, we have also used k-nearest neighbour classifiers (see Section 4.3).

Table 1 shows the identification performance of the C4.5 decision-tree classifier using bagging, for both types of feature vectors. The results show that the feature vectors of type 2 give better results than those of type 1, especially for the *mixed genera* data set. The same observations were found when using the k-nearest neighbour classifier (see Section 4.3). The main reason why type-1 feature vectors do not give good performances is the fact that the number of significant peaks in the first cluster (which is computed for type-1 features) is two, and therefore it is not sufficiently discriminating between the various diatom classes: most diatoms in the data sets considered here have two main peaks in their curvature signal (see Fig. 1). Note that diatoms with outlines similar to that shown in Fig. 3 do not appear in any of these data sets. Replacing the number of peaks by the extent and variance of the points with the highest curvature improves the discriminatory power of the feature set.

Table 1 Identification performance using the C4.5 decision tree classifier with bagging. The column '$\bar{x}$' contains the average number of errors; the column 'σ' contains the standard deviation of the number of errors; the columns 'min' and 'max' contain the minimum and maximum number of errors, respectively; the column 'perf.' contains the percentage (average with standard deviation) of samples identified correctly.

feature set	$\bar{x}$	σ	min	max	perf. (%)
Sellaphora pupula data set					
type-1	6.4	0.8	5	7	78.6 ± 0.3
type-2	5.2	1.1	4	8	82.6 ± 0.5
Mixed genera data set					
type-1	62.5	4.9	50	66	65.9 ± 2.5
type-2	29.5	4.9	23	39	84.0 ± 2.5

4.1 Comparison to the method of Leymarie and Levine

We conducted a second experiment in which we used the original method by Leymarie and Levine [25] to extract curvature features. As we mention in Section 2.3, the main differences between our method and the original method are: (i) we allow multiple nested peaks of the curvature to be associated with a certain section of the contour, and (ii) we do not split the boundary in convex and concave parts. The results of this experiment are shown in table 2. Note that only the type-2 feature vectors are computed. As it can be seen, the identification performances of the original method (i.e. no nested peaks allowed and boundary splitting performed – the rows marked with - + in table 2) for both data sets are smaller than those obtained with our modified method (table entries marked with + -). For the *Sellaphora* set, the identification performance when splitting is performed and nested peaks are allowed is worse than that obtained vice-versa (i.e. no splitting nor nested peaks). This can be explained by the fact that for all diatom shells from this data set, the curvature function possesses only two prominent peaks corresponding to the two convex endings of the shells, and in this case allowing for nested peaks is not very important. On the contrary, allowing for nested peaks seems important for the second set, because of the larger shape variations and shape complexities within this data set. For both data sets, the orig-

Table 2 Identification performance using the C4.5 decision tree classifier with bagging; results obtained using the type-2 feature vectors. The column '$\bar{x}$' contains the average number of errors; the column 'σ' contains the standard deviation of the number of errors; the columns 'min' and 'max' contain the minimum and maximum number of errors, respectively; the column 'perf.' contains the percentage (average with standard deviation) of samples identified correctly.

method						
nested	split	$\bar{x}$	σ	min	max	perf. (%)
Sellaphora pupula data set						
$-$	$-$	6.9	2.0	2	10	77.0 ± 0.9
$+$	$+$	7.8	2.3	3	11	74.0 ± 1.2
$-$	$+$	9.0	1.5	7	11	70.0 ± 0.7
$+$	$-$	5.2	1.1	4	8	82.6 ± 0.5
Mixed genera data set						
$-$	$-$	42.2	7.6	32	54	77.1 ± 5.3
$+$	$+$	36.2	4.5	30	43	80.4 ± 2.3
$-$	$+$	53.3	7.2	43	64	71.1 ± 5.0
$+$	$-$	29.5	4.9	23	39	84.0 ± 2.5

inal method performs worse than when any of the proposed modifications are made.

4.2 Comparison to other methods

A comparison between various techniques for contour-based feature extraction, in the context of diatom classification, is given in [42]. Relevant identification results for the *Mixed genera* set, taken from [42] are shown in Table 3.

Table 3 Identification performances of various feature sets using the C4.5 decision tree classifier with bagging; results obtained for the *Mixed genera* data set. The column 'size' contains the size of the feature set; the column '$\bar{x}$' contains the average number of errors; the column 'σ' contains the standard deviation of the number of errors; the columns 'min' and 'max' contain the minimum and maximum number of errors, respectively; the column 'perf.' contains the percentage (average with standard deviation) of samples identified correctly.

feature set	size	$\bar{x}$	σ	min	max	perf. (%)
moment inv.	11	59.4	4.9	50	66	67.9 ± 2.6
simple shape	7	46.3	3.8	40	53	75.0 ± 2.1
Fourier descr.	128	29.5	4.1	23	34	84.1 ± 2.2
Gabor features	99	84.2	5.8	77	94	54.5 ± 3.2
charact. prof.	40	46.2	4.6	41	57	75.0 ± 2.5

The first set of descriptors (moment inv.) contains the moment invariants proposed by Hu [19] and those by Flusser and Suk [13]. The second set consists of simple shape descriptors such as rectangularity, compactness, ellipticity, etc. [21]. The third set is given by the first 128 Fourier descriptors [46], while the fourth set of descriptors is comprises 99 Gabor features extracted using the method in [37]. The last set of descriptors (charact. prof.) is obtained by contour profiling based on dynamic ellipse fitting [8]. For detailed detailed information about these methods and their applications to diatom identification we refer to [8, 11, 37].

Note that the identification performance obtained using the morphological curvature scale space descriptor (Table 1) compares favorably with most results shown in Table 3. Only Fourier descriptors performed comparably (at the expense of large feature vectors), whereas the performances of all other methods are at least 9 percentage points smaller. Note that using larger feature vectors usually results in larger classifier training and testing times. Although training and testing decision trees are fast operations, other classifiers (e.g. neural networks) may need considerably larger training/testing times, especially for very large data sets. In [22] we show that building morphological curvature scale spaces is a fast operation ($O(N)$, with N the length of the contour), as opposed to Fourier descriptors which can be computed in $O(N \log N)$. Also, the results obtained using a different set of contour descriptors than those used here, extracted from the morphological curvature scale spaces are higher than those obtained using Fourier descriptors (see [22]). Surprisingly, Gabor descriptors and moment invariants performed worse than the method based on simple shape features.

4.3 Identification using the k-nearest neighbour classifier

The nearest-neighbour method assigns an unknown sample to the class of the most similar or 'nearest' sample point in the training set of data [17]. Nearest can be taken in the sense of smallest Euclidean distance in n-dimensional feature space. The k-nearest-neighbour technique is a more general version of the nearest neighbour approach in which an unknown sample is classified based on the majority vote of its k nearest neighbours, rather than on only its single nearest neighbour. For further details on the technique we refer to [17, 41].

One point not yet addressed is the choice of k. In a limited study, Enas and Choi give a rule $k \approx N^{2/8}$ or $k \approx N^{3/8}$ for the optimal value of k, where N is the size of the data set, cf. [41]. For the *Sellaphora* data set, we chose $k = 7$ corresponding to the latter approximation, and for the second set we chose a value intermediate between the two optimal estimates, i.e., $k = 10$. The identification results of the k-NN classifier, using the *leave-one-out* method for accuracy estimation, are given in Fig. 7, as a function of k. The maximum values used for k are given by the optimal estimates mentioned above.

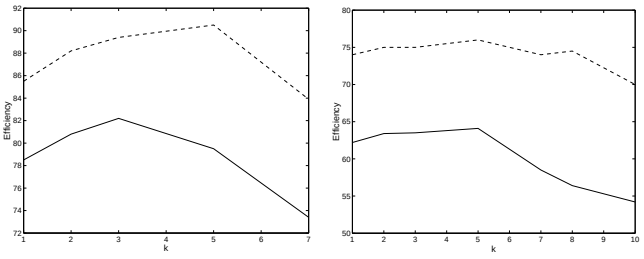

Fig. 7 The performance of the k-NN method using leave-one-out for accuracy estimation. *Left*: *Sellaphora pupula* data set, *right*: *Mixed genera* data set; *solid line*: type-1 feature vector, *dashed line*: type-2 feature vector.

As it can be seen from the figure, the performances are again higher using the type-2 feature vectors. For the *Mixed genera* data set the performance is around 75%, while for the *Sellaphora* data set the performance rises up to 90%, for values of k as large as 5. Note that in the case of the mixed genera set the maximum identification performance is smaller than that of the decision-tree classifier, whereas for the second set it is larger. In both cases the performance drops with increasing k, before the optimal values predicted by Enas and Choi are reached.

4.4 Implementation issues

A plain implementation of the curvature scale spaces (Section 2.3) results in a complexity of order $O(L \cdot N)$, where L is the largest scale and N is the number of values of the curvature function. This result is obtained when the opening transform is computed by a linear-time algorithm, insensitive to

the size of the structuring element, similar to that of Gil and Werman [15]. When L approaches N, the CPU time taken to construct the hat scale spaces is proportional with N^2. The curvature scale spaces can efficiently be implemented using Tarjan's Union-Find approach for maintaining disjoint sets. For more details, we refer to [28, 43].

The straightforward implementation of the mean-shift algorithm given in Section 3.2 results in a complexity of order $O(N^2)$, where N is the dimensionality of the data set to be clustered. An efficient implementation of this algorithm was presented by Comaniciu and Meer [9]. As discussed in the paper, the mean-shift algorithm clusters an n-dimensional data set by associating each point to a peak of the probability density of the data set. For each point, the mean-shift algorithm computes its associated peak by first defining a spherical window at the data point of radius h and computing the mean of the points that lie within the window. The algorithm then shifts the window to the mean and repeats until convergence, i.e. until the shifts are under some threshold. With each iteration the window shifts to a more densely populated portion of the data set until a peak is reached, where the data is equally distributed in the window. Upon finding a peak, each data point which lies at a distance smaller than h from some peak is associated with the cluster defined by that peak. This speedup is known as *basin of attraction* [9] and is based on the assumption that points that are within a distance smaller than h from the peak will converge with high probability to that peak. Further reduction in the computation time can be achieved by using efficient algorithms for range searching, employed for finding data points within the window.

Without using any these speedups, the CPU time spent to extract the feature vectors from all 781 contours of the mixed genera set, on a Pentium IV at 1.9 GHz, is 9 minutes.

5 Conclusions

We have proposed a method for diatom identification based on morphological curvature scale spaces, combined with unsupervised cluster analysis, for contour-based feature extraction. Before extracting the curvature measure, an adaptive smoothing method is used that filters the data according to a criterion depending on the signal-to-noise ratio. We have used a multi-scale approach to the analysis of 1-D signals, based on the work by Leymarie and Levine [25], proposed some modifications of their method to include nested features and developed a method for salient feature extraction. Unsupervised clustering of raw scale-space data turned out to be a useful method for this purpose. Experiments were carried out on two sets of diatom contour files, one containing demes of a species complex (*Sellaphora pupula*) and another data set with different diatom taxa. The C4.5 decision-tree classifier and the k-nearest neighbour classifier were used for diatom identification. Also, two types of feature vectors were used, denoted by Type-1 and Type-2 features. Type-1 was constructed by computing the number of peaks, mean curvature and variance for the first two clusters containing the scale-space features with the largest absolute curvatures, both for the top and bottom curvature scale spaces. Type-2 vectors were identical except that the number of peaks (valleys) was replaced by their extent. The results showed that identification accuracies of up to 84% are achievable using type-2 feature vectors, which is quite promising in view of the fact that the performance of human observers (who can make use of both contour *and* ornamentation information) is in the same range [23]. To elaborate, in the case of the *Sellaphora pupula* set, the identification performance using decision trees was slightly higher than the human experts' average of 82% (individual experts' results ranged from 60 to 98.3%) [6]. For the *mixed genera* data set, individual experts' identification rates ranged from 43 to 86.5% [6], which is an indication that the best computerized methods can compete with well-trained experts.

Type-1 feature vectors did not always give satisfactory results. We suspect that the number of peaks in the first two clusters containing features with high curvature is not sufficiently discriminating, because most diatoms in the data sets considered here have two main peaks in their curvature signal. Of course, it is possible that for other shape sets than those considered here type-1 features are better. It seems that the ratio between the support of each important feature and the length of the contour (extent), accompanied by the mean curvatures represent a robust yet discriminative pattern, which can be successfully used in diatom identification.

More fundamental research is needed on how to choose various parameters, which are now determined empirically. A comparison with diatom identification based on other contour extraction methods is needed as well. Such a comparison must include computational efficiency as one of the relevant criteria.

The present method is based on the analysis of contour features only. Work is in progress [23] to further improve identification accuracy with the help of diatomists by including morphological features of the diatom valve face to the feature sets which are input for the identification algorithm. The use of decision trees becomes very useful in this case, since features of different types (discrete, continuous, boolean) have to be combined. Finally, the explanatory power of decision trees allows human experts to interact with the classifier, for example through a graphical interface.

Acknowledgements This research is part of the ADIAC (Automatic Diatom Identification and Classification) project, funded by the European Commission MAST programme, contract MAS3-CT97-0122. Project home page: http://www.ualg.pt/adiac/.

References

1. Bangham, J.A., Chardaire, P., Pye, C.J., Ling, P.D.: Multiscale nonlinear decomposition: the sieve decomposition theorem. IEEE Trans. Pattern Anal. Machine Intell. **18**, 529–538 (1996)
2. Bangham, J.A., Ling, P.D., Harvey, R.: Scale-space from nonlinear filters. IEEE Trans. Pattern Anal. Machine Intell. **18**, 520–528 (1996)

3. Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Computer Vision and Image Understanding **64**(3), 377–389 (1996)
4. Breiman, L.: Bagging predictors. Machine Learning **24(2)**, 123–140 (1996)
5. du Buf, H., Bayer, M., Droop, S., Head, R., Juggins, S., Fischer, S., H. Bunke, M.W., Roerdink, J., Pech-Pacheco, J., Christobal, G., Shahbazkia, H., Ciobanu, A.: Diatom identification: A double challenge called ADIAC. In: Proceedings ICIAP, pp. 734–739. Venice (1999)
6. du Buf, H., Bayer, M.M.: ADIAC achievements and future work. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 14, pp. 289–298. World Scientific Publishing Co. , Singapore (2002)
7. Chen, M.H., Yan, P.F.: A multiscale approach based on morphological filtering. IEEE Trans. Pattern Anal. Machine Intell. **11**, 694–700 (1989)
8. Ciobanu, A., du Buf, H.: Identification by contour profiling and Legendre polynomials. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 9, pp. 167–185. World Scientific Publishing Co. , Singapore (2002)
9. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Machine Intell. **24**(5), 603–619 (2002)
10. Epanechnikov, V.A.: Nonparametric estimation of a multidimensioal probability density. Theor. Probab. Appl. **14**, 153–158 (1969)
11. Fischer, S., Bunke, H.: Identification using classical and new features in combination with decision tree ensembles. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 6, pp. 109–140. World Scientific Publishing Co. , Singapore (2002)
12. Fischer, S., Shahbazkia, H.R., Bunke, H.: Contour extraction. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 6, pp. 93–107. World Scientific Publishing Co. , Singapore (2002)
13. Flusser, J., Suk, T.: Pattern recognition by affine moment invariants. Pattern Recognition **26**, 167–174 (1993)
14. Fukunaga, K., Hostetler, L.D.: Estimation of the gradient of a density function with applications in pattern recognition. IEEE Trans. Inf. Th. **21**, 32–40 (1975)
15. Gil, J., Werman, M.: Computing 2-D min, median, and max filters. IEEE Trans. Pattern Anal. Machine Intell. **15**, 504–507 (1993)
16. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Reading, MA (1992)
17. Gose, E., Johnsonbaugh, R., Jost, S.: Pattern recognition and Image analysis. Prentice Hall PTR (1996)
18. Heijmans, H.J.A.M.: Morphological Image Operators, *Advances in Electronics and Electron Physics, Supplement*, vol. 25. Academic Press, New York (1994)
19. Hu, M.K.: Visual pattern recognition by moment invariants. IEEE Trans. Inf. Th. **8**, 179–187 (1962)
20. Jackway, P.T., Deriche, M.: Scale-space properties of the multiscale morphological dilation-erosion. IEEE Trans. Pattern Anal. Machine Intell. **18**, 38–51 (1996)
21. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ (1989)
22. Jalba, A.C., Wilkinson, M.H.F., Roerdink, J.B.T.M.: Shape representation and recognition through morphological curvature scale spaces. Tech. Rep. IWI 2003-10-01, Institute for Mathematics and Computing Science, University of Groningen (2003)
23. Kelly, M.G., Bayer, M.M., Hürlimann, J., Telford, R.J.: Human error and quality assurance in diatom analysis. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 5, pp. 75–91. World Scientific Publishing Co. , Singapore (2002)
24. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence, pp. 1137–1145 (1995)
25. Leymarie, F., Levine, M.D.: Curvature morphology. Tech. Rep. TR-CIM-88-26, Computer Vision and Robotics Laboratory, McGill University, Montreal, Quebec, Canada (1988)
26. Leyton, M.: Symmetry-curvature duality. Computer Vision Graphics and Image Processing **38**, 327–341 (1987)
27. Lindeberg, T.: Scale-space theory: A basic tool for analysing structures ar different scales. Journal of Applied Statistics **21**, 225–270 (1994)
28. Meijster, A., Wilkinson, M.H.F.: A comparison of algorithms for connected set openings and closings. IEEE Trans. Pattern Anal. Machine Intell. **24**(4), 484–494 (2002)
29. Mokhtarian, F., Abbasi, S., Kittler, J.: Efficient and robust retrieval by shape content through curvature scale space. In: A.W.M. Smeulders, R. Jain (eds.) Image DataBases and Multi-Media Search, pp. 51–58. World Scientific Publishing (1997)
30. Mokhtarian, F., Mackworth, A.K.: Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE Trans. Pattern Anal. Machine Intell. **8**, 34–43 (1986)
31. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE Trans. Pattern Anal. Machine Intell. **14**, 789–805 (1992)
32. Nacken, P.F.M.: Chamfer metrics in mathematical morphology. Journal of Mathematical Imaging and Vision **4**, 233–253 (1994)
33. Park, K.R., Lee, C.N.: Scale-space using mathematical morphology. IEEE Trans. Pattern Anal. Machine Intell. **18**, 1121–1126 (1996)
34. Pavlidis, T.: Algorithms for shape analysis of contours and waveforms. IEEE Trans. Pattern Anal. Machine Intell. **2**, 301–312 (1980)
35. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)
36. Round, F.E., Crawford, R.M., Mann, D.G.: The Diatoms: Biology and Morphology of the Genera. Cambridge University Press (1990)
37. Santos, L., du Buf, H.: Identification by Gabor features. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 10, pp. 187–220. World Scientific Publishing Co. , Singapore (2002)
38. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, New York (1982)
39. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
40. Stoermer, E.F., Smol, J.P. (eds.): The Diatoms: Applications for the Environmental and Earth Sciences. Cambridge University Press (1999)
41. Webb, A.: Statistical pattern recognition. Arnold (1999)
42. Westenberg, M.A., Roerdink, J.B.T.M.: Mixed-method identifications. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 12, pp. 245–257. World Scientific Publishing Co. , Singapore (2002)
43. Wilkinson, M.H.F., Jalba, A.C., Urbach, E.R., Roerdink, J.B.T.M.: Identification by mathematical morphology. In: J.M.H. Du Buf, M.M. Bayer (eds.) Automatic Diatom Identification, *Series in Machine Perception and Artificial Intelligence*, vol. 51, chap. 11, pp. 221–244. World Scientific Publishing Co. , Singapore (2002)
44. Wilkinson, M.H.F., Roerdink, J.B.T.M., Droop, S., Bayer, M.: Diatom contour analysis using morphological curvature scale spaces. In: Proc. 15th Intern. Conf. on Pattern Recognition (ICPR'2000), Barcelona, Spain, Sep. 3-7, pp. 656–659 (2000)
45. Witkin, A.P.: Scale-space filtering. In: Proc. Int. Joint Conf. Artificial Intell., pp. 1019–1022. Palo Alto, CA (1983)
46. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. IEEE Trans. Comput. **21**, 269–281 (1972)