Chapter 13

Diatom Identification with Mathematical Morphology

13.1. Introduction

Diatoms are single-cell algae with highly ornate silica shells. Each of the myriad species has a distinctive contour shape and internal texture (or ornamentation) which is used in identification at the (sub)species level. Diatoms occur in almost any aquatic or sufficiently humid environment, and each species is adapted to a restricted range of environmental conditions. This makes diatoms highly suitable as indicator species for environmental change. It also makes them suitable in various forensic investigations, e.g. in the case of drowning victims. Normally, population analysis is performed by human experts who laboriously scan microscope slides to locate diatom cells, and then compare their appearances under the microscope to that of reference images.

A few years ago, the EU-funded ADIAC (Automatic Diatom Identification and Classification [BAY 01, BUF 02]) project studied various methods to identify these diatoms automatically. Among the techniques developed were several from mathematical morphology, which proved to be among the fastest and most robust [WES 02, WIL 02]. Since then we have developed these multiscale techniques further [JAL 05, JAL 06, URB 07], and have achieved even higher identification performances. In this chapter we will present the techniques used to identify diatoms, first using morphological curvature scale spaces followed by a description of 2D shape-size pattern spectra. Finally, we will present the results using each technique separately and in combination.

Chapter written by Michael Wilkinson, Erik Urbach, Andre Jalba and Jos Roerdink.
After determining the curvature at each point along the boundary of the diatom, a morphological scale space is constructed by extracting peaks and valleys using a method derived from Leymarie and Levine [LEY 88]. Peaks at different scales are extracted using top-hat filters with flat structuring elements of increasing sizes. By performing repeated top-hat filters at increasing scales we can build a scale space. The result of each top-hat filter is compared to that of the preceding scale. If a feature present at the previous scale is not any larger in the current scale, its maximum height, average height, width and location are stored in a linked list structure. The peak is then subtracted from the original curve. The peak is then subtracted from the original curve. The difference between our method and that of Leymarie and Levine’s is that peaks may be nested in our case. A similar procedure is used for valleys, using bottom-hat operators. Unlike Leymarie and Levine [LEY 88], we do not first split the contour into concave and convex parts.

At the end of this procedure, we have two curvature scale spaces: one ‘bottom scale space’ of valleys and a ‘top scale space’ of peaks. In both scale spaces, each feature is represented by its start and end points on the contour, its extremum curvature value, its extremum position and its mean curvature value. 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 demonstrated in Figure 13.1.

Figure 13.1. Building curvature spaces: (a, d) binary images; (b, e) curvature plots; and (c, f) scale-space features displayed as blocks of the correct width and average height.
13.3. Scale-space feature extraction

Direct use of the scale-space data for the classification of shapes has several problems, in particular the fact that the pattern vectors of different shapes differ in length which is a problem for many classifiers.

To obtain a pattern vector of constant length we cluster the scale-space features using distribution-free cluster analysis. This ensures that we make no a priori assumptions about the number of clusters or the shape of their distribution. The mean-shift method we have chosen [FUK 75] is based on kernel density estimation. As a final step in the curvature feature extraction, we construct three 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. 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. For each cluster compute the mean curvature and the extent and variance of the points with the highest curvature.

– Type 3: take the $N_t$ and $N_b$ largest absolute curvature scale-space features from top- and bottom-hat scale spaces. For additional robustness, this is done for a series of different curvature signals, each obtained by Gaussian smoothing of the original curvature with an increasing sequence of standard deviations $\sigma_i$.

In addition, since curvature is a local attribute 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 is defined as the number of scale-space entries from both top and bottom curvature scale spaces. Hence, in the first two cases the size of the pattern vector is 14. In the case of type 3 pattern vectors, 4 values of $\sigma_i : 3.0, 6.0, 10.0, 16.0$ were used with $N_{t,i} : 10, 10, 5, 5$ and $N_{b,i} = 5$ for $i = 1, 2, 3, 4$. Two additional global shape measures were used: eccentricity and elongation [GON 87], yielding a pattern vector of length 66. The scale space is translation and rotation invariant, and scale invariance is obtained through normalization of the curvature measure data themselves.

13.4. 2D size-shape pattern spectra

In this section we describe how we compute an ornamentation feature vector based on a 2D size-shape pattern spectrum. We first recall the definition of a size distribution.
DEFINITION 13.1.– A binary size distribution is a set of operators \( \{\alpha_r\} \) with \( r \) from some totally ordered set \( \Lambda \) (usually \( \Lambda \subset \mathbb{R} \) or \( \mathbb{Z} \)), with the following three properties:

\[
\begin{align*}
\alpha_r(X) & \subset X, \\
X \subset Y & \Rightarrow \alpha_r(X) \subset \alpha_r(Y), \text{ and} \\
\alpha_r(\alpha_s(X)) & = \alpha_{\max(r,s)}(X)
\end{align*}
\]

for all \( r, s \in \Lambda \).

From equations (13.1) and (13.2) it can be seen that granulometries are ordered sets of openings, with some size parameters providing the order. Generalization to the grayscale case is straightforward [NAC 94, NAC 96, VIN 00].

In [URB 02, URB 07] scale-invariant operators are used to introduce shape-based granulometries. An operator \( \phi \) is scale-invariant if

\[
\phi(X_\lambda) = (\phi(X))_\lambda,
\]

for all \( \lambda > 0 \), where \( X_\lambda \) is a scaling of \( X \) by a factor \( \lambda \). A scale-invariant operator is sensitive to shape rather than to size. In practice we use shape operators which are scale, rotation and translation invariant. In [URB 07] it is shown that scale invariance and increasingness are not in general compatible. Shape operators are therefore not openings or closings. If we wish to define sets of shape filters analogous to size distributions, we must omit increasingness but include scale invariance.

DEFINITION 13.2.– A binary shape distribution is a set of operators \( \{\beta_r\} \) with \( r \) from some totally ordered set \( \Lambda \), with the following three properties:

\[
\begin{align*}
\beta_r(X) & \subset X, \\
\beta_r(X_\lambda) & = (\beta_r(X))_\lambda \text{ and} \\
\beta_r(\beta_s(X)) & = \beta_{\max(r,s)}(X)
\end{align*}
\]

for all \( r, s \in \Lambda \) and \( \lambda > 0 \).

A shape distribution consists of operators that are anti-extensive and idempotent but not increasing. These properties can be attained using attribute thinnings [BRE 96], also referred to as anti-extensive grain filters [HEI 99]. We do demand that the operators be scale-invariant, which is obtained through scale invariance of the attributes used [URB 07]. Extension to gray level is straightforward.

13.4.1. Shape and size pattern spectra

Pattern spectra [MAR 89] are essentially a histogram containing the number of pixels, or the amount of image detail over a range of size classes. Let \( r \) be the scale
parameter of a size distribution. The size class of \( x \) is then the smallest value of \( r \) for which \( x \notin \alpha_r(X) \). Shape pattern spectra can be defined analogously [URB 02]. If \( \{\alpha_r\} \) and \( \{\beta_r\} \) are size and shape distributions, respectively, the corresponding spectra \( s_{\alpha}(X) \) and \( s_{\beta}(X) \) are defined as

\[
(s_{\alpha}(X))(u) = -\frac{dA(\alpha_r(X))}{dr}\bigg|_{r=u} \tag{13.8}
\]

and

\[
(s_{\beta}(X))(u) = -\frac{dA(\beta_r(X))}{dr}\bigg|_{r=u} \tag{13.9}
\]

where \( A(X) \) is the Lebesgue measure in \( \mathbb{R}^n \) (= area if \( n = 2 \)). In the grayscale case, the pattern spectrum is usually defined in terms of the sum or integral of the gray levels over the image domain as a function of \( r \):

\[
(s_{\alpha}(f))(u) = \frac{d}{dr} \int (\alpha_r(f))(x)dx\bigg|_{r=u} , \tag{13.10}
\]

and likewise for \( s_{\beta}(f) \).

In the discrete case we can compute a pattern spectrum by repeatedly filtering an image by each \( \alpha_r \), in ascending order of \( r \). After each filter step, the sum of gray levels \( S_r \) of the resulting image \( \alpha_r(f) \) is computed. The pattern spectrum value at \( r \) is computed by subtracting \( S_r \) from \( S_{r^-} \), with \( r^- \) the scale immediately preceding \( r \). In practice, faster methods for computing pattern spectra can be used [BRE 96, MEI 02, NAC 94, NAC 96]; see also section 13.4.3.

13.4.2. Attribute thinnings

Attribute thinnings as defined by Breen and Jones [BRE 96] can provide shape granulometries [URB 07]. Binary attribute thinnings are based on binary connectivity openings. The binary connectivity opening \( \Gamma_x(X) \) of \( X \) at point \( x \) is the connected component or grain of \( X \) containing \( x \) if \( x \in X \), and \( \emptyset \) otherwise. A trivial thinning \( \Phi_T \), which uses a non-increasing criterion \( T \), is then used to accept or reject each grain. The trivial thinning \( \Phi_T \) of set \( C \) with criterion \( T \) is \( C \) if \( C \) satisfies \( T \), and \( \emptyset \) otherwise. The binary attribute thinning is defined as follows.

**Definition 13.3.**— The binary attribute thinning \( \Phi_T^X \) of set \( X \) with criterion \( T \) is given by

\[
\Phi_T^X = \bigcup_{x \in X} \Phi_T(\Gamma_x(X)). \tag{13.11}
\]
If both the connectivity opening $\Gamma_x$ and criterion $T$ are scale invariant, so are $\Phi_T$ and $\Phi^T_T$ (for proofs see [URB 07]). In the following we assume that the connectivity opening is scale invariant (which is true for 4- and 8-connectivity). Let $T(C)$ be of the form

$$\tau(C) \geq r, \quad r \in \Lambda,$$

(13.12)

with $\tau$ some scale-invariant attribute of the connected set $C$ and $\Lambda$ a totally ordered set. As an example, $\tau(C)$ might be one of the moment invariants of Hu [HU 62]. Let the attribute thinnings using these $T$ be denoted as $\Phi^r_T$. We have

$$\Phi^r_T(\Phi^s_T(X)) = \Phi^{\max(r,s)}_T(X),$$

(13.13)

proving that $\{\Phi^r_T\}$ is a shape distribution because attribute thinnings are anti-extensive and scale invariance is provided by the scale invariance of $\tau(C)$. Grayscale generalizations of these filters are discussed in [BRE 96, SAL 98].

### 13.4.3. Computing 2D shape-size pattern spectra

To implement grayscale versions of attribute thinnings, it is best to use a data structure called the max-tree by Salembier et al. [SAL 98] and component tree by some others [JON 99, NAJ 06] (see Chapter 7). It is a tree structure which encodes the connected components of every threshold set of a grayscale image, their nesting relationships and their attribute values. Once an image is represented by such a tree, filtering reduces to removing nodes from the tree and reconstructing the resulting image. Each node $C^k_h$ of the max-tree corresponds to the $k$th connected foreground component or peak component $P^k_h$ of the threshold set at gray level $h$ in the image. An example is depicted in Figure 13.2.

We now show how to compute a 2D pattern spectrum from a max-tree. Let $\{\alpha_r\}$ be a size distribution with $r$ from some finite index set $\Lambda_r$ and $\{\beta_s\}$ a shape distribution with $s$ from some index set $\Lambda_s$. With these distributions, we can compute a discrete 2D size-shape pattern spectrum similar to [GHO 98]. Assume that we store this 2D pattern spectrum in a 2D array $S$, in which $S(r,s)$ contains the sum of gray levels of those features in the image which fall within the size class between $r^-$ and $r$ and shape class between $s^-$ and $s$, where $r^-$ and $s^-$ are members of $\Lambda_r$ and $\Lambda_s$ immediately preceding $r$ and $s$, respectively. We can then compute a 2D pattern spectrum from an image as follows:

- set all elements of the array $S$ to zero;
- compute a max-tree according to the algorithm in [SAL 98];
- as the max-tree is built, compute the area $A(P^k_h)$ and moment of inertia $I(P^k_h)$ of each node;
Figure 13.2. A 1D signal showing its peak components $P_k^h$, the attributes (length and some fictitious shape attribute), the corresponding max-tree and the spectrum. In the latter, the horizontal axis is the shape axis and the size axis is vertical. The upper left corner is bin $(1, 1)$. Both axes are linear in this case. Each bin contains the sum of the product of peak-component area and the gray-level difference from its parent in the max-tree. Note that the root node does not contribute, because its signal strength is zero.

- for each node $C_k^h$:
  - compute the size class $r$ from the area $A(P_k^h)$;
  - compute the shape class $s$ from $\frac{I(P_k^h)}{A^2(P_k^h)}$, which is equal to the first moment invariant of Hu [HU 62];
  - compute the gray-level difference $\delta_h$ between the current node and its parent;
  and
  - add the product of $\delta_h$ and $A(P_k^h)$ to $S(r, s)$.

It remains to be specified how to transform size and shape attributes to size and shape classes. Let $v$ be the attribute value, $c$ the class, $N_c$ the number of classes and $D_0$ and $D_1$ the lower and upper bound of the range of interest of the attribute value, respectively. The following heuristic was used to compute the class $c$:

$$c = \left(\log_2(v) - \log_2(D_0)\right) \left(\log_2(D_1) - \log_2(D_0)\right)^{-1} N_c. \quad (13.14)$$

In our dataset the size range is from $D_0 = 1$ pixels to $D_1 = 1550^2$ pixels. For the shape parameter $D_0 = 1/2\pi$ (theoretical minimum) and $D_1 = 52$. The number
of classes is 20 for size and 15 for shape (optimum found experimentally). Finally, we computed the complete ornamentation feature vector by concatenating the pattern spectrum of the image with that of the inverted image. The first is a pattern spectrum of the bright details, the second one of the dark details. Two variants of these feature vectors were computed: normalized and non-normalized. In the first case we divide the value of each bin of the spectrum by the sum of all bins; in the second case the raw spectra were used.

13.5. Datasets

The classification results were computed on the ADIAC database [BUF 02] for 781 images of diatoms. Each of the 37 species was represented by at least 20 images. These images were obtained by bright-field microscopy with a $1018 \times 1018$ or $1312 \times 1312$ pixel CCD-camera. The images were cropped to about $600 \times 400$ pixels. Each shows one diatom cell. The size range of the diatoms themselves is quite large, varying from 5 to 200 $\mu$m in length. For each grayscale image, a binary image was available which masked all non-diatom pixels in the image. The watershed-based segmentation method is given in [JAL 04]. Pattern spectra computed were restricted to the mask region. The images of the ADIAC database are obtainable from http://www.ualg.pt/adiac/pubdat/pubdat.html. Figure 13.3 shows a pattern spectrum corresponding to a diatom image.

![Diatom Image and Pattern Spectrum](image)

Figure 13.3. A diatom image with the corresponding pattern spectrum. The vertical axis shows the area and the horizontal the first moment invariant of Hu of image features in each bin; brightness indicates the power in each bin. One selected bin in each spectrum and the corresponding image details are highlighted by a hatch pattern.

13.6. Results

Table 13.1 shows the results of classification using the C4.5 decision-tree-based classifier, with bagging to provide robustness and 10-fold cross-validation to estimate
the error. As can be seen, the type 3 feature vector for the curvature scale space and the binned, non-normalized feature vector for the 2D pattern spectra yield the best classification results. The performance of the combination of the best contour method and the best ornamentation method (93.9 ± 3.5) is clearly better than the earlier result of 89.6 ± 1.9 presented in [WIL 02]. This difference is due to improved feature extraction from both the morphological curvature scale space and the 2D pattern spectrum. It also compares well with the results of the combination of all diatom features obtained in the ADIAC project: 96.9 ± 1.2 [WES 02]. The great advantage of using only the morphological methods is their faster speed compared to some of the other methods used in the ADIAC project. Computing all 781 pattern vectors of the ADIAC data base required 17.4 seconds CPU time (including I/O) on a 2.9 GHz Pentium 4, as opposed to several hours for the complete ADIAC pattern vector of length 329.

<table>
<thead>
<tr>
<th>Method</th>
<th>Size</th>
<th>( \mu ) (%)</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curvature scale space</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type 1</td>
<td>14</td>
<td>65.9</td>
<td>2.5</td>
</tr>
<tr>
<td>Type 2</td>
<td>14</td>
<td>84.0</td>
<td>2.5</td>
</tr>
<tr>
<td>Type 3</td>
<td>66</td>
<td>91.3</td>
<td>5.0</td>
</tr>
<tr>
<td>2D pattern spectra</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moments</td>
<td>98</td>
<td>74.8</td>
<td>3.6</td>
</tr>
<tr>
<td>Binned</td>
<td>600</td>
<td>91.1</td>
<td>1.6</td>
</tr>
<tr>
<td>Binned + norm.</td>
<td>600</td>
<td>79.5</td>
<td>3.3</td>
</tr>
<tr>
<td>Combined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binned + type 3</td>
<td>666</td>
<td>93.9</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Table 13.1. Classification performance on a set of 781 diatom images; size refers to the feature-vector length

13.7. Conclusions

Multiscale mathematical morphology provides a computationally efficient and powerful means for the identification of diatoms. Because no assumptions on shape or size of the cells are made, the methods are directly applicable to other classes of organisms. Scale invariance yields robustness to differences in magnification (i.e. type of microscope) and rotation invariance eliminates the need to manually align each cell to a standard orientation. Automatic alignment is possible in elongated cells, but becomes unstable in more-or-less circular cells or the trilobate types shown in Figure 13.1. Both methods described have also been tested in other applications such as shape retrieval on the MPEG-7 database for the curvature scale space (78.8% performance [JAL 06]) and texture classification on the Brodatz database (96.5% performance for the 2D pattern spectra [URB 07]).