

# Connected Shape-Size Pattern Spectra for Rotation and Scale-Invariant Classification of Gray-Scale Images

Erik R. Urbach, *Student Member, IEEE*, Jos B.T.M. Roerdink, *Senior Member, IEEE*, and Michael H.F. Wilkinson, *Senior Member, IEEE*

**Abstract**—In this paper, we describe a multiscale and multishape morphological method for pattern-based analysis and classification of gray-scale images using connected operators. Compared with existing methods, which use structuring elements, our method has three advantages. First, in our method, the time needed for computing pattern spectra does not depend on the number of scales or shapes used, i.e., the computation time is independent of the dimensions of the pattern spectrum. Second, size and strict shape attributes can be computed, which we use for the construction of joint 2D shape-size pattern spectra. Third, our method is significantly less sensitive to noise and is rotation-invariant. Although rotation invariance can also be approximated by methods using structuring elements at different angles, this tends to be computationally intensive. The classification performance of these methods is discussed using four image sets: Brodatz, COIL-20, COIL-100, and diatoms. The new method obtains better or equal classification performance to the best competitor with a 5 to 9-fold speed gain.

**Index Terms**—Mathematical morphology, connected filters, scale spaces, multiscale analysis, shape filters, rotation-invariance, diatoms, Brodatz textures, COIL-100 object library.

## 1 INTRODUCTION

PATTERN spectra [17], [30] are commonly used tools for image analysis and classification, which can be computed using a technique from mathematical morphology [18] known as granulometries [3], [30] (for a review of granulometries, see [40]). Intuitively, a size granulometry can be considered as a set of sieves of different grades, each allowing details of certain size classes to pass. More formally, a size granulometry consists of an ordered set of operators, each of which converts an image to a new image in which features smaller than a particular size are absent.

Granulometries are computed using filters with structuring elements (S.E.) or using connected filters [11], [20], [27], [28], [33], [41]. Connected filters are shape preserving because they never introduce new edges in images. Various algorithms exist to compute pattern spectra efficiently, both for structural [9], [32], [34], [35], [39] and connectivity-based methods [3], [19], [20].

Breen and Jones [3] proposed a type of granulometry based on attribute openings. These allow the use of many size attributes other than width, such as area, length of the diagonal of the minimum enclosing rectangle, moment of inertia, etc., to define the “grades” of the morphological sieves. They also put forward the idea of attribute thinnings which allow image filtering based on shape, rather than size

attributes. An attribute that depends only on shape information is called a strict shape attribute and the corresponding granulometry a *shape granulometry* [38]. These allow extraction of pattern spectra based on shapes, rather than sizes of details.

In a preliminary version of this study [37], gray-scale multivariate pattern spectra computed from connected operators, notably the joint shape-size pattern spectra, were discussed together with the subtractive filtering rule for filtering Max-trees. The time needed to compute pattern spectra using this method does not depend on the number of scales or shapes used. The main contributions of the present paper are: 1) an extensive discussion of the theory behind rotation-invariant and strict shape attributes and operators, including new proofs of various theoretical properties, and 2) an extensive comparison of the classification performance of several variations of our method and existing methods using structuring elements (S.E. methods), similar to [1], [5], using four different image sets: Brodatz texture database, COIL-20, and COIL-100 object databases, and a diatom database [6]. The new method obtains better or equal performance to the best competitor with a 5 to 9-fold speed gain and is considerably less sensitive to noise than any of the existing methods used here.

In the following section, the theoretical background is described, which is used to develop the new theory and proofs of Section 3. The algorithms for computing size and shape attributes, the computation of our joint shape-size pattern spectrum, and some existing methods using structuring elements are discussed in Section 4. The classification performance and computing time of these S.E. methods are compared with our novel method using experiments on four image sets in Section 5. The last section is devoted to some

• The authors are with the Institute for Mathematics and Computing Science, University of Groningen, PO Box 800, 9700 AV Groningen, The Netherlands.

E-mail: erikurbach@yahoo.com, {j.b.t.m.roerdink, m.h.f.wilkinson}@rug.nl.

Manuscript received 4 Nov. 2005; revised 15 May 2006; accepted 28 June 2006; published online 13 Dec. 2006.

Recommended for acceptance by R. Klette.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0601-1105.

discussion and conclusions. For easy reference, an appendix is included which contains a summary of symbols and notational conventions used in this paper.

## 2 BACKGROUND

In the following, binary images  $X$  and  $Y$  are defined as subsets of the image domain  $E \subseteq \mathbb{R}^n$  or  $\mathbb{Z}^n$  (usually  $n = 2$ ), and gray-scale images are mappings from  $E$  to  $\mathbb{R}$  or  $\mathbb{Z}$ . The power set  $\mathcal{P}(E)$  is the set of all subsets of  $E$ . As a convention, operators producing binary images are denoted here by a capital Greek letter; the corresponding lowercase letter is used to denote the gray-scale version. When  $X \subseteq E$ , we denote by  $X^x$  the translation of  $X$  over the vector  $x \in E$ .

The theory of granulometries and connected filters is presented only very briefly here. For more detail, the reader is referred to [3], [18], [27], [30], [40].

### 2.1 Size Pattern Spectra

**Definition 1.** A binary size granulometry is a set of operators  $\{\Gamma_r\}$  with  $r$  from some totally ordered set  $\Lambda$  (usually,  $\Lambda \subseteq \mathbb{R}$  or  $\mathbb{Z}$ ) with the following three properties:

$$\Gamma_r(X) \subseteq X, \quad (1)$$

$$X \subseteq Y \Rightarrow \Gamma_r(X) \subseteq \Gamma_r(Y), \quad (2)$$

$$\Gamma_r(\Gamma_s(X)) = \Gamma_{\max(r,s)}(X), \quad (3)$$

for all  $r, s \in \Lambda$ .

Since (1) and (2) define  $\Gamma_r$  as antiextensive and increasing, respectively, and (3) implies idempotence, it can be seen that size granulometries are sets of openings. Generalization to the gray-scale case  $\{\gamma_r\}$  is straightforward and has been discussed in, e.g., [21], [22], [40].

Size pattern spectra were introduced by Maragos [17]. Essentially, they are a histogram containing the number of pixels, or the amount of image detail, over a range of size classes. If  $r$  is the scale parameter of a size granulometry and  $X$  a binary image, the size class of a point  $x \in X$  is the smallest value of  $r$  for which  $x \notin \Gamma_r(X)$ . The pattern spectrum  $s_\Gamma(X)$  obtained by applying the size granulometry  $\{\Gamma_r\}$  to a binary image  $X$  is defined as

$$(s_\Gamma(X))(u) = - \left. \frac{d\xi(\Gamma_r(X))}{dr} \right|_{r=u}, \quad (4)$$

where  $\xi$  denotes some measure, for which the common Lebesgue measure  $A(X)$  in  $\mathbb{R}^n$  will be used here, which is just the area if  $n = 2$ . In the gray-scale case, the pattern spectrum is usually defined by using the integral  $A(f)$  (sum of the gray levels) of  $f$  over the image domain as the measure.

In the discrete case, we can compute a pattern spectrum of a gray-scale image by repeatedly filtering by each  $\gamma_r$ , in ascending order of  $r$ . After each filter step, the sum of gray levels  $S_r$  of the resulting image  $\gamma_r(f)$  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 [3], [19], [21], [22], see also Section 4.2. These methods do not apply each  $\gamma_r$  separately. However, for methods using structuring elements in gray-scale, this is usually unavoidable.

### 2.2 Grains, Flat Zones, and Connected Filters

By a *grain* of a binary image  $X$ , we mean a connected component of  $X$ . The  $i$ th grain of  $X$  is denoted by  $C_i^X$ , where  $i$  runs over some index set  $I^X$ .

A *flat zone*  $L_h(f)$  of a gray-scale image  $f$  is a connected component of the level set  $\mathcal{V}_h(f) = \{x \in E | f(x) = h\}$ . At each gray level, there may be multiple flat zones, which are denoted by  $L_h^k(f)$ , with  $k$  some index variable. Similarly, a *peak component*  $P_h^k(f)$ , where  $k$  runs over some index set  $I_h^f$ , is defined as the  $k$ th connected component or *grain* of the threshold set  $\mathcal{T}_h(f)$  of image  $f$ , which is defined as

$$\mathcal{T}_h(f) = \{x \in E | f(x) \geq h\}. \quad (5)$$

(We will often suppress the image argument  $f$  of flat zones and peak components when no confusion arises.) Connected filters only merge flat zones, or change their gray level, but never split nor deform them. This means no new edges are introduced. More formally, for any image  $f$ , the flat zones define a partition  $\{\Upsilon_i\}$  of the image domain  $E$ . A filter  $\gamma$  is connected if the partition  $\{\Omega_j\}$  defined by  $\gamma(f)$  is always coarser than  $\{\Upsilon_i\}$ , for any image  $f$  [28]. By coarser, we mean that, for any  $\Upsilon_i$ , there exists a  $\Omega_j$  such that  $\Upsilon_i \subseteq \Omega_j$  [28].

### 2.3 Attribute Operators

Binary attribute openings and thinnings are based on binary connected openings. The binary connected opening  $\Gamma_x(X)$  of  $X$  at point  $x \in E$  yields the grain (connected component) of  $X$  containing  $x$  if  $x \in X$ , and  $\emptyset$  otherwise. Breen and Jones [3] use the concept of a nonincreasing criterion  $T$  to accept or reject grains. Formally, a criterion  $T$  is a mapping of sets to Booleans, i.e.,  $T : \mathcal{P}(E) \rightarrow \{\text{false}, \text{true}\}$ . A criterion  $T$  is increasing if the implication  $(T(C) \Rightarrow T(D))$  holds whenever  $C \subseteq D$ .

The trivial thinning  $\Phi_T$  of a grain  $C$  with criterion  $T$  is just the set  $C$  if  $C$  satisfies  $T$ , and the empty set otherwise. Furthermore,  $\Phi_T(\emptyset) = \emptyset$ .

**Definition 2.** The binary attribute thinning  $\Phi^T$  of a set  $X$  with criterion  $T$  is given by

$$\Phi^T(X) = \bigcup_{x \in X} \Phi_T(\Gamma_x(X)). \quad (6)$$

Though it was called an attribute thinning in [3], a better name is perhaps an antiextensive, nonincreasing grain filter [11]. This, however, is rather cumbersome, so we will stick to the name attribute thinning. What (6) says is that the attribute thinning is equivalent to performing a trivial thinning on all grains in the image, i.e., removing all grains which do not meet the criterion. Note also that (6) defines a binary attribute *opening* when the criterion  $T$  is increasing.

Various gray-scale generalizations of these filters are compared in [16], [27]. An efficient algorithm for computing these filters based on a so-called Max-tree of flat zones in the image was put forward in [27]. The filtering is reduced to different methods of "pruning" this tree structure. In the following, this algorithm and various thinnings based on different pruning strategies are discussed.

### 2.4 The Max-Tree

The *Max-tree* representation [27] separates the filtering process into three steps: construction, filtering, and restitution. It is a tree where the nodes represent sets of flat zones. The Max-tree node  $N_h^k$  consists of the subset of  $P_h^k$  with gray

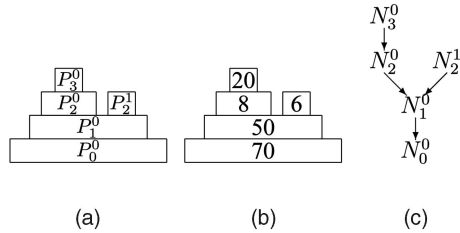


Fig. 1. (a) The peak components of a gray-level image  $X$ , (b) the corresponding fictitious attributes, and (c) the Max-tree.

level  $h$ . The root node represents the set of pixels belonging to the background, which is the set of pixels with the lowest intensity in the image. Each node has a pointer to its parent, i.e., the nodes corresponding to the components with the highest intensity are the leaves (see Fig. 1). Conversely, a tree in which the leaves correspond to the minima is called a Min-tree and can be used for attribute closings or thickenings. During the construction phase, the Max-tree is built from the flat zones of the image.

After this, the tree is processed during the filtering phase. Based on the attribute value  $\tau(P_h^k)$  of a node  $N_h^k$ , the algorithm takes a decision on whether to preserve or to remove it. This is done by associating a criterion  $T$  to the attribute function  $\tau$ .

**Definition 3.** We call a criterion  $T$  on subsets of  $E$  associated to  $\tau$  when  $T(X) = \rho(\tau(X))$  with  $\rho: \mathbb{R} \rightarrow \{\text{false}, \text{true}\}$  for all  $X \subseteq E$ .

For example, we often use the criterion  $T(C) = (\tau(C) < r)$ , where  $r$  is a scalar value.

Two classes of strategies exist: 1) pruning strategies, which remove all descendants of  $N_h^k$  if  $N_h^k$  is removed, and 2) nonpruning strategies in which the parent pointers of children of  $N_h^k$  are updated to point at the oldest “surviving” ancestor of  $N_h^k$ . If  $T$  is nonincreasing, such as  $T(C) = (P(C) < r)$ , where  $P$  denotes perimeter, pruning strategies must either remove nodes which meet  $T$  or keep nodes which do not. Salembier et al. [27] describes four different rules to filter the tree: the *Min*, *Max*, *Viterbi*, and *Direct* rule. The first three are pruning strategies. In addition to these, we describe a new, nonpruning strategy: the *Subtractive* rule. The decisions of these rules are as follows:

- *Min*. A node  $N_h^k$  is removed if  $\tau(P_h^k) < r$  or if one of its ancestors is removed.
- *Max*. A node  $N_h^k$  is removed if  $\tau(P_h^k) < r$  and, for all its children,  $N_{h'}^j$ ,  $\tau(P_{h'}^j) < r$  holds.
- *Viterbi*. The removal and preservation of nodes is considered as an optimization problem. For each leaf node, the path with the lowest cost to the root node is taken, where a cost is assigned to each transition. For details, see [27].
- *Direct*. A node  $N_h^k$  is removed if  $\tau(P_h^k) < r$ ; its pixels are lowered in gray level to the highest ancestor which meets  $T$ , and while its descendants are unaffected.
- *Subtractive*. As above, but the descendants are lowered by the same amount as  $N_h^k$  itself.

Let  $\{C_i^{T_h(f)}\}$  be the set of grains contained in any of the threshold sets of some digital image  $f$ , with  $\mathcal{T}_h(f)$  the threshold set at gray level  $h \in \{h_{\min} + 1, h_{\min} + 2, \dots, h_{\max}\}$ , and, for every fixed  $h$ ,  $i$  is from some index set  $I_h^f$ . These

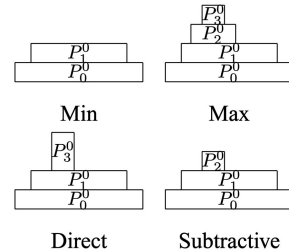


Fig. 2. Result after filtering the image in Fig. 1 with four different decision rules, using a value of 10 as the attribute threshold.

grains correspond to the peak components of  $f$ , that is,  $C_i^{T_h(f)} = P_h^i(f)$ . Note that, for any two sets  $X, Y \in \{C_i^{T_h(f)}\}$  either  $X \subseteq Y$ ,  $Y \subseteq X$ , or  $X \cap Y = \emptyset$ . Furthermore, let  $\chi$  denote the characteristic function for a binary image  $X$ :

$$(\chi(X))(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases}$$

We define the subtractive filter formally as follows:

**Definition 4.** The gray-scale attribute thinning by the subtractive rule is defined by

$$\phi^T(f) = h_{\min} + \sum_{h=h_{\min}+1}^{h_{\max}} \sum_{i \in I_h^f} \chi(\Phi_T(C_i^{T_h(f)})). \quad (8)$$

What this equation says is that the result  $\phi^T(f)(x)$  of the attribute thinning at every pixel  $x$  can be obtained by initializing it with a value  $h_{\min}$ , considering all gray levels in turn, and, at each gray level  $h$ , adding a value of 1 for every grain  $C_i^{T_h(f)}$  which satisfies  $T$  and contains  $x$ .

Fig. 1 shows the peak components of a 1D gray-scale image, their (fictitious) attribute values, and the corresponding Max-tree. The results of applying the Min, Max, Direct, and Subtractive rules on this image with  $r = 10$  are shown in Fig. 2.

### 3 SHAPE-SIZE PATTERN SPECTRA

Below, granulometries and pattern spectra based on strict shape attributes will be defined. These are used to develop the novel joint 2D shape-size pattern spectrum.

#### 3.1 Shape Operators

Let the scaling of set  $X$  by a scalar factor  $t \in \mathbb{R}$  be denoted by  $tX$ , which is

$$tX = \{x \in \mathbb{R}^n | t^{-1}x \in X\}. \quad (9)$$

Likewise, a scaling  $tf$  of a gray-scale image  $f$  is defined as

$$(tf)(x) = f(t^{-1}x) \quad \forall t^{-1}x \in E. \quad (10)$$

A binary operator  $\Phi$  or gray-scale operator  $\phi$  is said to be *scale-invariant* if

$$\Phi(tX) = t(\Phi(X)) \quad \text{or} \quad \phi(tf) = t(\phi(f)) \quad (11)$$

for all  $t > 0$ . A scale-invariant operator is therefore sensitive to shape rather than to size. If an operator is scale, rotation and translation invariant, we call it a *shape operator*; if it is also idempotent, it is a *shape filter*. Note that, if  $E \subseteq \mathbb{Z}^n$ , scale

invariance can only be attained approximately because in general,  $t^{-1}x \notin \mathbb{Z}^n$ , so one has to apply rounding to the nearest element of  $\mathbb{Z}^n$ .

The *hit-or-miss transform*  $X \otimes \mathbf{B}$ , with  $\mathbf{B} = (B_1, B_2)$  a pair of nonoverlapping structuring elements, can be used to define the *thinning*  $X \circ \mathbf{B}$ :

$$X \otimes \mathbf{B} = (X \ominus B_1) \cap (X^c \ominus B_2), \quad (12)$$

$$X \circ \mathbf{B} = X \setminus (X \otimes \mathbf{B}), \quad (13)$$

in which  $X^c$  is the complement of  $X$ ,  $X \setminus Y$  is the set difference, and  $X \ominus B$  is the erosion of  $X$  with structuring element (S.E.)  $B$ .

Thinnings using structuring elements cannot be used as shape filters, in part because they are not, in general, idempotent but more importantly because they are not scale invariant:

$$t(X \circ \mathbf{B}) = tX \setminus (tX \otimes t\mathbf{B}) \neq tX \circ \mathbf{B} \quad (14)$$

in which  $t\mathbf{B} = (tB_1, tB_2)$ .

**Definition 5 (Grain Preservation).** A binary image transformation  $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  is called grain preserving if each grain of  $X \subseteq E$  is mapped uniquely to a single grain of  $\Psi(X)$  (and vice versa).

This means that the set of grains  $\{C_i^X\}$  of  $X$  and  $\{C_i^{\Psi(X)}\}$  of  $\Psi(X)$  can be indexed using the same index set  $I$ , that is,

$$\Psi(C_i^X) = C_i^{\Psi(X)}, \quad i \in I, \quad (15)$$

$$\Psi(\emptyset) = \emptyset. \quad (16)$$

**Definition 6.** A function  $\Sigma : \mathcal{P}(E) \rightarrow \mathbb{D}$ , with arbitrary range  $\mathbb{D}$ , is called invariant under a mapping  $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  if  $\Sigma(X) = \Sigma(\Psi(X))$  for all  $X \subseteq E$ .

Examples of such functions  $\Sigma$  are the attribute function  $\tau : \mathcal{P}(E) \rightarrow \mathbb{R}$  or the criterion function  $T : \mathcal{P}(E) \rightarrow \{\text{false}, \text{true}\}$ .

**Proposition 1 (Attribute versus Operator Invariance).** If the attribute  $\tau : \mathcal{P}(E) \rightarrow \mathbb{R}$  is invariant under a grain-preserving image transformation  $\Psi$ , so are  $\Phi_T$  and  $\Phi^T$ , where  $T$  is a criterion associated to  $\tau$ .

**Proof.** Let  $\tau : \mathcal{P}(E) \rightarrow \mathbb{R}$  be some attribute function and let  $C \subseteq X$  be a grain. If  $\tau$  is invariant under  $\Psi$ , i.e.,

$$\tau(C) = \tau(\Psi(C)), \quad (17)$$

and  $\rho : \mathbb{R} \rightarrow \{\text{false}, \text{true}\}$  be any mapping from real values to Booleans, then a criterion  $T$  of the form  $T(C) = \rho(\tau(C))$  satisfies

$$T(C) = \rho(\tau(C)) = \rho(\tau(\Psi(C))) = T(\Psi(C)), \quad (18)$$

so  $T$  is also invariant under  $\Psi$ .

Furthermore, the trivial thinning  $\Phi_T$  is also invariant under  $\Psi$ , i.e.,

$$\Phi_T(\Psi(C)) = \Psi(\Phi_T(C)). \quad (19)$$

To prove this, note that

$$\Psi(\Phi_T(C)) = \begin{cases} \Psi(C) & \text{if } T(C) \\ \emptyset & \text{otherwise} \end{cases} \quad (20a)-(20b)$$

through the definition of trivial thinnings and (16), whereas

$$\Phi_T(\Psi(C)) = \begin{cases} \Psi(C) & \text{if } T(\Psi(C)) \\ \emptyset & \text{otherwise} \end{cases} \quad (21a-21b)$$

can be rewritten as (see (18)),

$$\Phi_T(\Psi(C)) = \begin{cases} \Psi(C) & \text{if } T(C) \\ \emptyset & \text{otherwise,} \end{cases} \quad (22a-22b)$$

which, combined with (20), proves (19).

The invariance of  $\Phi_T$  to  $\Psi$  also implies invariance of the attribute thinning  $\Phi^T$  of (6), which can also be written as

$$\Phi^T(X) = \bigcup_{i \in I} \Phi_T(C_i^X). \quad (23)$$

To prove this, we first note that

$$\Psi\left(\bigcup_{i \in I} \Phi_T(C_i^X)\right) = \bigcup_{i \in I} \Psi(\Phi_T(C_i^X)). \quad (24)$$

To show this, let  $I' \subseteq I$  be the index set of all grains of  $X$  which satisfy  $T$  and let  $X' = \sum_{i \in I'} C_i^X$  so that  $C_i^{X'} = C_i^X$  for  $i \in I'$ . Then,

$$\begin{aligned} \Psi\left(\bigcup_{i \in I} \Phi_T(C_i^X)\right) &= \Psi\left(\bigcup_{i \in I'} C_i^X\right) = \Psi(X') \\ &= \bigcup_{i \in I'} C_i^{\Psi(X')} = \bigcup_{i \in I'} \Psi(C_i^{X'}) \\ &= \bigcup_{i \in I'} \Psi(C_i^X) = \bigcup_{i \in I'} \Psi(\Phi_T(C_i^X)), \end{aligned}$$

where we used that  $\Psi$  is grain-preserving in the sense of (15). Using this formula, and the invariance of  $\Phi_T$ , we find

$$\Psi(\Phi^T(X)) = \Psi\left(\bigcup_{i \in I} \Phi_T(C^X)\right) \quad (25)$$

$$= \bigcup_{i \in I} \Psi(\Phi_T(C_i^X)) = \bigcup_{i \in I} \Phi_T(\Psi(C_i^X)) \quad (26)$$

$$= \bigcup_{i \in I} \Phi_T(C_i^{\Psi(X)}) = \Phi^T(\Psi(X)), \quad (27)$$

which proves the invariance of  $\Phi^T$  to  $\Psi$ .  $\square$

For example, if  $\Psi$  is a scaling and the connectivity used is scale invariant, i.e., scaling is grain preserving in the sense of (15), scale invariance of the attribute implies scale invariance of the attribute filter. Examples of connectivities for which this holds are four and eight connectivity in the digital case, or path connectivity in the continuous case. Note that scale invariance in the discrete case cannot always be ensured for objects that contain some thin paths when the resolution is decreased.

### 3.2 Shape Granulometries

Below, we use the theory of size granulometries to define shape granulometries, which are invariant to scale. To exclude sensitivity to size, the operators used can generally not be increasing, as we will be show below. Therefore, if we wish to define sets of shape filters analogous to size granulometries, we must omit property (2), but include scale invariance. This motivates the following definition:

**Definition 7.** A binary shape granulometry is a set of operators  $\{\Phi_r\}$  with  $r$  from some totally ordered set  $\Lambda$ , with the following three properties:

$$\Phi_r(X) \subseteq X, \quad (28)$$

$$\Phi_r(tX) = t(\Phi_r(X)), \quad (29)$$

$$\Phi_r(\Phi_s(X)) = \Phi_{\max(r,s)}(X), \quad (30)$$

for all  $r, s \in \Lambda$  and  $t > 0$ .

Thus, a shape granulometry consists of operators which are antiextensive, idempotent, and scale invariant. As in the case of size granulometries, translation and rotation invariance are not demanded in this definition, though these operators are often desirable. Extension to gray-level operators is straightforward.

We now show that antiextensive shape operators are generally not increasing. As a first example, suppose we have two binary images,  $X$  and  $Y$ , with  $X \subset Y$ . Furthermore, suppose there exists a  $t$  such that  $tY \subset X$ . Let  $\Phi$  be a binary shape operator such that  $\Phi(X) = \emptyset$  and  $\Phi(Y) \neq \emptyset$ . Then, according to (11),  $\Phi(tY) \neq \emptyset$ , which means  $\Phi(tY) \not\subseteq \Phi(X)$  and  $\Phi$  is not increasing. A more general result is given in the next proposition.

**Proposition 2 (Scale Invariance versus Increasingness).**

Any antiextensive and scale and translation-invariant operator  $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$  is nonincreasing if there exists a bounded set  $X$ , for which  $\Psi(X)$  is nonempty, and there exists a ball  $B_r^x$  of radius  $r$  and centered on  $x \in E$  which is completely contained in the difference set  $X \setminus \Psi(X)$ .

**Proof.** We first note that  $B_r^x \subset X$ , but  $B_r^x \cap \Psi(X) = \emptyset$ . Because  $X$  is bounded, there exists a ball  $B_R$  such that  $X \subseteq B_R$ . Let  $Y = (\frac{r}{R}X)^x$ . Then,

$$Y \subseteq \left(\frac{r}{R}B_R\right)^x = B_r^x \subseteq X.$$

Since  $\Psi$  is scale and translation invariant,

$$\Psi(Y) = \Psi\left(\left(\frac{r}{R}X\right)^x\right) = \left(\frac{r}{R}\Psi(X)\right)^x,$$

so  $\Psi(Y)$  is not empty since  $\Psi(X)$  is not empty. Because  $\Psi$  is antiextensive,

$$\Psi(Y) \subseteq Y \subseteq B_r^x.$$

Since  $B_r^x$  is disjoint from  $\Psi(X)$  and  $\Psi(Y)$  is not empty,  $\Psi(Y) \not\subseteq \Psi(X)$ . Thus, we have  $Y \subseteq X$  but  $\Psi(Y) \not\subseteq \Psi(X)$ , which proves that  $\Psi$  is not increasing.  $\square$

In the digital case, we always have bounded images and any meaningful antiextensive operator will remove a ball of some finite size (if only a pixel), and leave some part of the image intact. As exceptions, there are two straightforward, but not useful, antiextensive operators which do combine scale and translation invariance with increasingness, and these are the identity operator  $I$  for which  $I(X) = X$  for any  $X \in \mathcal{P}(E)$  and "null" operator  $O$  defined by  $O(X) = \emptyset$  for any  $X \in \mathcal{P}(E)$ .

Note that, in the extensive case, the situation is different. Let  $\Psi$  be a Minkowski addition with a ball  $B_{r(X)}$  as a structuring element, in which  $r(X)$  is a linearly increasing translation-invariant measure of  $X$ , i.e.,

$$Y \subseteq X \Rightarrow r(Y) \leq r(X), r(tX) = tr(X), r(X^h) = r(X). \quad (32)$$

Examples of such measures are the erosion width of the largest grain of  $X$  or a fixed fraction of the square root of the area of  $X$ . Let  $Y \subseteq X$ , with  $r(Y) = \epsilon r(X)$  with  $0 < \epsilon \leq 1$ . We now have

$$\Psi(X) = X \oplus B_{r(X)} = \bigcup_{x \in X} B_{r(X)}^x \quad (33)$$

and

$$\Psi(Y) = Y \oplus B_{r(Y)} = \bigcup_{x \in Y} B_{r(Y)}^x. \quad (34)$$

Because  $Y \subseteq X$  and  $B_{r(Y)}^x = B_{\epsilon r(X)}^x \subseteq B_{r(X)}^x$  for any  $x$ , it is clear that  $\Psi(Y) \subseteq \Psi(X)$ , proving increasingness. Translation invariance is obvious as well, and scale invariance is easily shown by observing that  $tB_{r(X)} = B_{tr(X)} = B_{r(tX)}$  and, therefore,

$$\begin{aligned} t\Psi(X) &= tX \oplus tB_{r(X)} \\ &= tX \oplus B_{r(tX)} = \Psi(tX). \end{aligned} \quad (35)$$

So, nontrivial operators exist which are both increasing and satisfying the requirements of extensive shape operators, i.e., scale and translation invariance.

### 3.3 Shape Pattern Spectra

Shape pattern spectra can be defined in a similar way [38] as the size pattern spectra. If  $r$  is the shape parameter of a shape granulometry and  $X$  a binary image, the shape class of  $x \in X$  is the smallest value of  $r$  for which  $x \notin \Phi_r(X)$ . The pattern spectrum  $s_\Phi(X)$  obtained by applying the shape granulometry  $\{\Phi_r\}$  to a binary image  $X$  with measure  $\xi$  is defined as

$$(s_\Phi(X))(u) = - \left. \frac{d\xi(\Phi_r(X))}{dr} \right|_{r=u}. \quad (36)$$

Assume the criterion  $T(C)$  can be written as  $\tau(C) \geq r$ ,  $r \in \Lambda$ , with  $\tau$  some scale-invariant attribute of the grain  $C$ . Let the corresponding attribute thinnings be denoted by  $\Phi_r^\tau$ . It can readily be shown that

$$\Phi_r^\tau(\Phi_s^\tau(X)) = \Phi_{\max(r,s)}^0(X). \quad (37)$$

Therefore,  $\{\Phi_r^\tau\}$  is a shape granulometry, since attribute thinnings are antiextensive, and scale invariance is provided by the scale invariance of  $\tau(C)$ , see Proposition 1. From the definition of pattern spectra, it is easy to obtain the following result:

**Proposition 3 (Invariance of Pattern Spectra).** Any pattern spectrum  $(s(X))(u)$  computed by a granulometry  $\{\Phi_r^\tau\}$  and measure  $\xi$ , which are both invariant under some grain-preserving image transformation  $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ , is invariant to  $\Psi$ , i.e.,  $(s(X))(u) = (s(\Psi(X)))(u)$ .

**Proof.** This is obvious from (27) and the invariance of  $\xi$  under  $\Psi$ .  $\square$

Examples of scale-invariant shape attributes are many. Obvious choices are  $P^2/A$ , with  $P$  the perimeter, and  $A$  the area of a grain, or  $I/A^2$ , which is the ratio of the moment of inertia  $I$  to the square of the area. These attributes attain minimal values for discs and increase as the objects become more elongated. For computational reasons, the latter shape attribute is used, because both moment of inertia and area

can easily and accurately be computed incrementally, whereas that is more difficult with perimeter. A fuller discussion of shape attributes is given in [3], [27].

### 3.4 Image Decomposition

To decompose an image  $f$  into its constituent components based on shape rather than size, using a gray-scale attribute thinning  $\phi_r^\tau$ , we wish to split the image content into: 1) an image  $\phi_r^\tau(f)$  containing *only* peak components  $P_h^k$  for which  $\tau(P_h^k) \geq r$  and 2) a difference image  $f - \phi_r^\tau(f)$  consisting of those peak components  $P_h^k$  for which  $\tau(P_h^k) < r$  holds. These requirements now mean that all peak components of  $\phi_r^\tau(f)$  meet  $\tau$ , and all peak components of  $f - \phi_r^\tau(f)$  do not. More formally:

**Definition 8 (Gray-Scale Shape Decomposition).** *A gray-scale attribute thinning  $\phi_r^\tau$  is a gray-scale shape decomposition if, for any gray-scale image  $f$  and for all  $h$ , the following two properties hold:*

$$\Phi_r^\tau(\mathcal{T}_h(\phi_r^\tau(f))) = \mathcal{T}_h(\phi_r^\tau(f)), \quad (38)$$

$$\Phi_r^\tau(\mathcal{T}_h(f - \phi_r^\tau(f))) = \emptyset. \quad (39)$$

We will now show that, of the five filtering rules described in Section 2.4, the subtractive rule alone can be used for shape decomposition.

**Proposition 4 (Pruning Strategies).** *No gray-scale attribute thinnings based on Max-tree pruning strategies are gray-scale shape decompositions.*

**Proof.** Consider an image  $f$  with just three nested peak components  $P_3^1 \subset P_2^1 \subset P_1^1$  at gray levels 3, 2, and 1, respectively. Furthermore, let  $\tau(P_3^1) \geq r$ ,  $\tau(P_2^1) < r$ , and  $\tau(P_1^1) \geq r$ . No pruning strategy can simultaneously retain  $P_3^1$  and  $P_1^1$  while removing  $P_2^1$ . This means it cannot satisfy both (38) and (39).  $\square$

**Proposition 5 (Direct Rule).** *No gray-scale attribute thinnings based on the (nonpruning) Max-tree direct rule are shape decompositions.*

**Proof.** Using the direct rule and the same image  $f$  as used above, the difference  $f - \phi_r^\tau(f)$  will consist of a zero background with one or more connected regions at gray level 1, consisting of those pixels of  $P_2^1$  which have gray level 2, i.e., the members of  $N_2^1$  (which need not be connected). In general, a peak component  $P_h^k$  of this difference image may satisfy  $T(P_h^k) = (\tau(P_h^k) \geq r)$ , in which case, it violates (39). This would, for example, be the case for an elongation criterion  $T$  that preserves elongated components and removes everything else, with  $P_2^1$  a  $3 \times 3$  square and  $P_3^1$  a  $3 \times 2$  rectangle. The difference image  $f - \phi_r^\tau(f)$  will contain a  $3 \times 1$  elongated component that is preserved by  $T$ .  $\square$

**Proposition 6 (Nonpruning Strategies).** *Gray-scale attribute thinnings based on nonpruning strategies are gray-scale shape decompositions if they are based on the subtractive rule.*

**Proof.** Any image  $f$  can be reconstructed completely from the set of grains  $\{C_i^{\mathcal{T}_h(f)}\}$  as follows:

$$f = h_{\min} + \sum_{h=h_{\min}+1}^{h_{\max}} \sum_{i \in \mathcal{I}_h^f} \chi(C_i^{\mathcal{T}_h(f)}), \quad (40)$$

where  $\chi$  is the characteristic function as defined in formula (4). Because the grains belonging to different threshold sets are nested within each other, the definition of the subtractive filter, (8), implies that the grains of threshold sets of  $\phi^T(f)$  are those elements of  $\{C_i^{\mathcal{T}_h(f)}\}$  that satisfy criterion  $T$ , which means (38) holds. Subtracting (8) from (40) yields

$$f - \phi^T(f) = \sum_{h=h_{\min}+1}^{h_{\max}} \sum_{i \in \mathcal{I}_h^f} \chi(\Phi_{-T}(C_i^{\mathcal{T}_h(f)})) \quad (41)$$

with  $\neg T$  the logical negation of criterion  $T$ . So, for all grains of the threshold set  $\mathcal{T}_h(f - \phi^T(f))$ , the criterion  $T$  is false. Therefore, (39) holds.  $\square$

Because the subtractive rule alone satisfies both (38) and (39), it is used in the following algorithm for 2D shape pattern spectra.

## 4 ALGORITHM

### 4.1 Computation of the Attributes

Salembier et al.'s Max-tree algorithm [27] was used to compute the pattern spectra. For each node  $N_h^k$  of the Max-tree corresponding to peak component  $P_h^k$ , two attributes are computed: The area  $A$  as the size attribute and the elongation measure  $I/A^2$ , defined as the ratio between the moment of inertia and the square of the area, as the strict shape attribute. The Max-tree algorithm as described in [27] already includes the computation of the area: Each time a pixel is added to a node, the corresponding area attribute value is incremented by one.

The elongation measure used is identical to the first moment invariant  $I_1 = \eta_{20} + \eta_{02}$  of Hu [12] and is computed using the following four kinds of moments [10] on gray-level image  $f$ :

$$\text{Moments: } m_{pq} = \iint_{\mathbb{R}^2} x^p y^q f(x, y) dx dy, \quad (42)$$

$$\text{Central moments: } \mu_{pq} = \iint_{\mathbb{R}^2} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy, \quad (43)$$

$$\text{Normalized moments: } n_{pq} = \frac{m_{pq}}{m_{00}^\gamma}, \quad (44)$$

$$\text{Normalized central moments: } \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad (45)$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}, \quad \text{and} \quad \gamma = \frac{p+q}{2} + 1.$$

As with [3], [27], gray-level information within each node  $N_h^k$  is ignored, and the moments of the characteristic function  $\chi(P_h^k)$  of the corresponding peak component are computed.

If we consider  $I_1 = \eta_{20} + \eta_{02}$  more closely, we find that  $I_1$  can be computed using only the following central moments:  $\mu_{00}$ ,  $\mu_{20}$ , and  $\mu_{02}$ , which can be defined in terms of moments  $m_{pq}$  [10] as follows:

$$\mu_{00} = m_{00}, \quad (46)$$

$$\mu_{20} = m_{20} - \bar{x} m_{10}, \quad (47)$$

$$\mu_{02} = m_{02} - \bar{y} m_{01}. \quad (48)$$

From (46), (47), and (48), it is easy to see that  $I_1$  can be computed using the following moments:  $m_{00}, m_{10}, m_{01}, m_{20}$ , and  $m_{02}$ , where  $m_{00}, m_{10}$ , and  $m_{01}$  are also needed for the computation of  $\bar{x}$  and  $\bar{y}$ . Thus, our elongation measure can be computed incrementally during the construction of the Max-tree, i.e., each time a pixel is added to a node, only a constant number of operations are required for the computation of the attribute. For example, let  $t_{pq}$  be the variable used to compute  $m_{pq}$ ; then, for each pixel  $p(x, y)$  processed,  $t_{pq}$  is incremented by  $x^p y^q$ .

## 4.2 Binned 2D Shape-Size Pattern Spectra

Suppose we have a gray-scale size granulometry  $\{\gamma_r\}$  with  $r$  from some finite index set  $\Lambda_r$  and a gray-scale shape granulometry  $\{\phi_s\}$  with  $s$  from some finite index set  $\Lambda_s$ . With these granulometries, we can compute a 2D pattern spectrum similar to that of Batman and Dougherty [1] and Ghosh and Chanda [8]. We can then store this 2D pattern spectrum, which contains both shape and size information in a 2D array  $S$ , in which  $S[r, s]$  contains the sum of gray levels of those features in the image which fall in the size class between  $r^-$  and  $r$  and shape class between  $s^-$  and  $s$ . Here,  $r^-$  and  $s^-$  are the members of  $\Lambda_r$  and  $\Lambda_s$  immediately preceding  $r$  and  $s$ , respectively. The computation of a 2D pattern spectrum  $S$  using binning is shown in Algorithm 4.1.

**Algorithm 4.1** Computing 2D shape-size binned pattern spectrum  $S$  using Max-tree with  $N_a$  shape and  $N_b$  size classes.

- Set all  $N_a \times N_b$  elements of the array  $S$  to zero.
- Compute a Max-tree according to the algorithm in [27].
- As the Max-tree is built, compute the area  $A(P_h^k)$  and moment of inertia  $I(P_h^k)$  of each node  $N_h^k$  corresponding to grain  $P_h^k$ .
- For each node  $N_h^k$ :
  - Compute the size class  $r$  from the area  $A(P_h^k)$ .
  - Compute the shape class  $s$  from  $I(P_h^k)/A^2(P_h^k)$ .
  - Compute the gray-level difference  $\delta_h$  between the current node and its parent.
  - Add the product of  $\delta_h$  and  $A(P_h^k)$  to  $S[r, s]$ .

What remains to be specified is how to transform size and shape attributes to size and shape bins. 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 bounds of the range of interest of the attribute value, respectively. Intuitively, one would consider the size of, for example, dots, in terms of their width or diameter, which can be expressed as logarithmic functions of the area. For this reason, and because most of the information about image details is concentrated in the smaller peak components, we wish to have finer bins for low attribute values than for high ones. Therefore, we use the following heuristic to compute the class  $c$ :

$$c = \left\lfloor \frac{\log_2(v) - \log_2(D_0)}{\log_2(D_1) - \log_2(D_0)} N_c \right\rfloor, \quad (49)$$

where  $\lfloor \dots \rfloor$  denotes the floor function. The resulting pattern spectrum is then mapped in lexicographic order into a 1D vector. The values for the parameters  $D_0$ ,  $D_1$ , and  $N_c$

depend on the application and will therefore be discussed in Section 5.

## 4.3 Computing Moments from 2D Shape-Size Pattern Spectra

Instead of binning to reduce the size of pattern spectra, moments are also often used for that purpose [29]. Usually, this is done by computing the first few moments from a discrete (finely binned) pattern spectrum. Since floating point values are used for the attributes in our Max-tree-based method, we decided to implement a close approximation of a continuous pattern spectrum by computing these moments directly using the nondiscrete attribute values as shown in Algorithm 4.2.

**Algorithm 4.2** Computing  $M \times N$  shape-size pattern spectrum moments  $S$  using Max-tree.

- Set all  $M \times N$  elements of the array  $m_{pq}$  to zero.
- Compute a Max-tree according to the algorithm in [27].
- For each node  $N_h^k$  corresponding to grain  $P_h^k$ :
  - Compute area  $x = A(P_h^k)$ .
  - Compute moment of inertia  $y = I(P_h^k)/A^2(P_h^k)$ .
  - Compute the gray-level difference  $\delta_h$  between the current node and its parent.
  - Update each  $m_{pq}$ , where  $f(x, y)$  is the product of  $\delta_h$  and  $A(P_h^k)$ .
- Set each element  $S[p, q]$  according to (50) using the computed values of  $m_{pq}$  as defined in (42).

This method creates a 2D array  $S$  using the moment functions. The normalized central moments are the most appropriate to use, however,  $\eta_{00} = 1$  and  $\eta_{10} = \mu_{10} = \eta_{01} = \mu_{01} = 0$ , hence, they contain no information. Instead of  $\eta_{00}$ , we will use the nonnormalized central moment  $\mu_{00} = m_{00}$  and, instead of  $\eta_{10}$  and  $\eta_{01}$ , the normalized geometric moments  $n_{01}$  and  $n_{10}$ , respectively. Therefore, this method computes  $S$  as follows:

$$S[p, q] = \begin{cases} m_{00} & \text{if } p = 0 \text{ and } q = 0, \\ n_{01} & \text{if } p = 0 \text{ and } q = 1, \\ n_{10} & \text{if } p = 1 \text{ and } q = 0, \\ \eta_{pq} & \text{otherwise.} \end{cases} \quad (50)$$

Note that  $S$  is invariant under translation and rotation of the input image since the computation of  $S$  can be modeled as applying the moments function to a pattern spectrum, which, in this case, is translation and rotation-invariant according to Proposition 3.

## 4.4 Extracting Feature Vectors

To obtain features, array  $S$  is mapped lexicographically into a vector  $\mathbf{v}_n$ . Now,  $\mathbf{v}_n$  contains information about bright patterns in the image. To obtain information about the dark patterns in a vector  $\mathbf{v}_i$ , antisize and antishape granulometries are required. Instead of designing new algorithms for the computation of pattern spectra based on these antisize and antishape granulometries, these were implemented as the computation of pattern spectra of inverted images. Since many images are classified using both the bright and the dark patterns, we decided to use  $\mathbf{v}_r$ , which is the concatenation of  $\mathbf{v}_n$  and  $\mathbf{v}_i$ , as a combined feature vector.

TABLE 1  
Image Libraries

image library	images	classes	images/class	image size
Diatoms	781	37	$\geq 20$	203x139 (min.) 1190x559 (max.)
Brodatz	2048	32	64	64x64
COIL-20	1440	20	72	128x128
COIL-100	7200	100	72	128x128

#### 4.5 Pattern Spectra Based on Structuring Elements

Three existing types of size granulometries based on openings  $X \circ B_i$  [30] with structuring elements  $B_i$  [1], [5] were implemented for comparison purposes. These methods compute pattern spectra using sequences  $\{B_i\}$  of S.E.s. Each of these sequences is constructed by defining some basic shape (generator)  $B_1$  and computing increasingly larger versions  $B_i$  of it by dilation  $B_i = B_{i-1} \oplus B_1$ . The following S.E. sequences were used: horizontal linear, vertical linear, +45-degree (positive diagonal) linear, -45-degree (negative diagonal) linear, flat-top (square), cone-shaped, and (approximately) circle-shaped. While circular S.E.s can be used efficiently for binary images [25], this becomes computationally expensive for the gray-scale case. So, for the circular case, a polygonal approximation [34], [35] was used, since we found that a direct implementation of a circular S.E. was more than 50 times slower than the approximation, while no improvement in classification performance was measured.

Efficient algorithms [32], [39] with a computing time invariant to the size of the structuring element were used for all openings and closings with structuring elements. The following existing S.E. methods were used to compare with our proposed Max-tree based approach:

- *S.E. univariate (UV)* [5]: For each of the seven S.E. sequences  $B_i$ , a separate 1D (univariate) pattern spectrum was computed using filter  $\gamma_t(f) = f \circ tB_i$  for  $t > 0$ , after which these seven pattern spectra were joined together. The pattern spectra  $(s_{B_i}(f))(u)$  were computed by:

$$(s_{B_i}(f))(u) = \left. \frac{d[1 - A(\gamma_t(f))/A(f)]}{dt} \right|_{t=u}, \quad (51)$$

where  $A(f)$  is the sum over all gray levels of  $f$ .

- *S.E. Euclidean* [1]: A single pattern spectrum was computed using (51) from five S.E. sequences (the seven described above minus the square and the circle to avoid redundancy). For  $t > 0$  and  $R_0 \leq t \leq R_1$ , a filter  $\gamma_t(f) = \bigcup_{i=1}^n f \circ tB_i$  was used to construct an Euclidean granulometry.
- *S.E. bivariate (BV)*: [1] A multivariate pattern spectrum was computed using the following granulometry:  $\gamma_t(f) = \bigcup_{i=1}^n f \circ t_i B_i$ , where  $\mathbf{t} = \{t_1, \dots, t_n\}$ . For computational reasons, only the horizontal and the vertical linear S.E. sequences were used, resulting in a pattern spectrum  $S[h, v] = 1 - A(\gamma_{\mathbf{t}_{h,v}}(f))/A(f)$ , with  $\mathbf{t}_{h,v} = \{t_h, t_v\}$ ,  $1 < h \leq H$ , and  $1 < v \leq V$ . For each element of the pattern spectrum, two values are computed and stored in the feature vector: its horizontal and its vertical derivative.

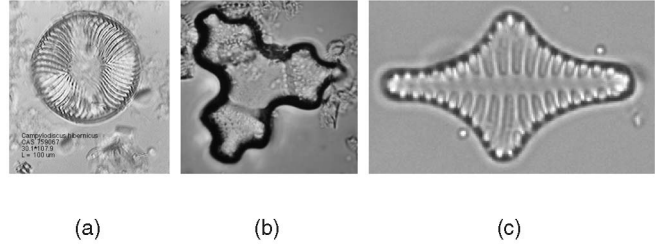


Fig. 3. Some examples of diatom valves.

Similar to the Max-tree method, two versions exist for the S.E. UV and the S.E. Euclidean methods: 1) one using a coarsely binned pattern spectrum directly and 2) one using moments of a finely binned pattern spectrum. For computational reasons, the moments version was not attempted for the S.E. BV method.

Since these S.E. methods use openings, they can be used to gather information about bright image structures. To extract information about dark image structures, closings are needed. Similar to the case of the Max-tree method described earlier, two pattern spectra are computed, one using openings and one using closings, which are then combined into a single feature vector of size  $N_c$ .

## 5 EXPERIMENTS

### 5.1 Image Libraries

The classification performances of the Max-tree methods were compared with existing S.E. methods using the following image libraries, see also Table 1. All images were converted to 8-bit gray-scale.

- *Diatoms*: Diatoms [36] are unicellular algae with highly ornate silica shells, which consist of two halves called *valves*. Examples of diatom valves are shown in Fig. 3. Both the shapes of these shells and the textures or *ornamentation* are distinctive characteristics of individual taxa (species or varieties). The methods described here compute the feature vectors from the ornamentation only.

The classification results were computed on a set of 781 images of diatoms from the ADIAC database [6], all of which are aligned manually by experts to the same orientation. Each of the 37 distinct taxa was represented by at least 20 images. These images were obtained using bright-field microscopy with a  $1,018 \times 1,018$  or  $1,312 \times 1,312$  pixel CCD-camera. The images were cropped to a region of about  $600 \times 400$  pixels, on average, surrounding the diatom cells, which vary from 5 to 200  $\mu\text{m}$  in length. For each gray-scale image, a binary image was available in which a single foreground component indicates which pixels belong to the diatom valve. The segmentation method is given in [14]. For diatoms, all methods described in this paper compute pattern spectra that are restricted to the valve pixels. The images of the ADIAC database are obtainable from <http://www.ualg.pt/adiac/pubdat/pubdat.html>.

- *COIL*: The Columbia Object Image Library (COIL) exists in three versions on the Web <http://www1.cs.columbia.edu/CAVE/research/softlib/>: COIL-20 (unprocessed), COIL-20 (processed), and COIL-100. All three sets were created by taking



72 pictures of each object on a turntable, i.e., for every 5 degrees of rotation, one picture was taken. The COIL-100 and COIL-20 (processed) libraries contain images which are histogram stretched and resized to  $128 \times 128$  pixels, while preserving the aspect ratio. The COIL-20 (unprocessed) set contains only five classes, and was therefore not used.

- *Brodatz*: The Brodatz texture database can be obtained from [http://www.ee.oulu.fi/research/imag/texture/image\\_data](http://www.ee.oulu.fi/research/imag/texture/image_data). Using the same experimental set-up as [15] images of  $256 \times 256$  pixels were divided into 16 disjoint squares of size  $64 \times 64$ , and, for each of these smaller images, three additional versions were created by one of the following transformations: 1) 90 degrees rotation, 2) scaling the  $45 \times 45$  subimage in the center to  $64 \times 64$ , or 3) a combination of 1) or 2).

## 5.2 Classification and Statistical Analysis

The classification technique we used is based on decision trees built with the C4.5 algorithm [24]. We do not use single decision trees for classification, but use *bootstrap aggregation* or *bagging* [4] to increase the stability of the decision tree classifier. The procedure is as follows:

1. Randomly divide the data representing  $N$  images into a training set ( $N_{\text{train}}$  images) and a test set ( $N_{\text{test}}$  images), ensuring that the test set contains, for each class, at least 25 percent of the images.
2. Construct 25 new training sets by drawing a number of  $N_{\text{train}}$  images from the initial training set randomly *with replacement* (bootstrapping), and build decision trees with each of them.
3. Evaluate each of the decision trees by computing its accuracy  $acc_i$  and take a majority vote on the outcomes of each tree (aggregation).
4. Repeat Steps 2 and 3 of this procedure  $N_{\text{runs}} = 10$  times to obtain the classification performance and an error estimate (cross validation).

The classification performance can be computed as follows: Let  $V$  be the space of unlabeled instances (images), and  $Y$  the space of classes. For each run  $i \in 1, 2, \dots, N_{\text{runs}}$  (in our case,  $N_{\text{runs}} = 10$ ), we obtain a bagging classifier  $C_i$  which maps unlabeled instances  $v_k \in V$  to  $Y$ . Let the test set in run  $i$  be denoted as  $D_{\text{test}}^i \subseteq V \times Y$ . The elements of  $D_{\text{test}}^i$  are denoted by  $\langle v_k, y_k \rangle$ , with  $v_k$  the image and  $y_k$  the class it belongs to (ground truth). The accuracy of a single bagging classifier  $C_i$  on a test set is computed as

$$acc_i = \frac{1}{N_{\text{test}}} \sum_{\langle v_k, y_k \rangle \in D_{\text{test}}^i} \text{Match}(C_i(v_k), y_k), \quad (52)$$

in which  $\text{Match}(y_k, y_l) = 1$  if  $y_k = y_l$ , and 0 otherwise. The classification performance is the average accuracy over the 10 runs.

## 5.3 Parameters

The parameters  $D_0$  and  $D_1$  for the Max-tree-based methods, as defined in (49), were determined for both size and shape classes. We decided that image details with minimum attribute values should also contribute to the pattern spectrum. Thus,  $D_0 = 1$  for the size class and, for the shape class,  $D_0 = 1/2\pi$ . The upper bound of the range of interest was tested for all image sets with different values. For the image

TABLE 2  
Classification Performance on a Set of 781 Diatom Images, Size Denotes the Feature-Vector Length

Method	Size	Performance	
		$\mu$ (%)	$\sigma$
S.E. UV moments	182	76.9	2.7
S.E. UV binned	224	89.5	2.4
S.E. Euclidean moments	24	55.1	2.9
S.E. Euclidean binned	28	77.5	3.6
S.E. BV binned	676	93.8	2.8
Max-tree moments	98	74.8	3.6
Max-tree binned	600	91.1	1.6
Max-tree binned norm.	600	79.5	3.3

sets diatoms, Brodatz, COIL-20, and COIL-100, the best classification performances were achieved using upper bounds for the size class of, respectively, 1,  $550^2$ ,  $128^2$ ,  $128^2$ , and  $436^2$  and for the shape class, respectively: 52, 70, 53, and 44.

For the S.E. methods, the openings and closings are computed for S.E. lengths between  $R_0$  and  $R_1$ . The smallest S.E. used was set to  $R_0 = 7$  pixels for all image sets. The best classification performance was reached with  $R_1 = 447$  for S.E. UV and S.E. Euclidean and  $R_1 = 127$  for S.E. BV. The same lower and upper bounds were used for the methods based on pattern spectrum moments as their binned counterparts described above.

The optimal value for the parameter controlling the size  $N_c$  of the feature vector was determined for all methods by computing the classification performance of pattern spectra for increasing values of  $N_c$  until the classification performance was maximal. We found that, except for the shortest feature vectors (i.e., less than five classes per attribute/S.E. shape), the length of the feature vector does not influence the classification performance significantly. Although a very large value could be chosen for  $N_c$ , this would be undesirable, since the computational cost of the classifier increases with the length of the feature vector and at some point the ‘‘curse of dimensionality’’ [13] is to be expected.

## 5.4 Results

In this section, we present the experimental results. First, we compute the classification performance on the diatom set to determine the best methods. Then, for the best two methods found, we determine the classification performance on all data sets. Next, we consider the behavior under image rotation and the influence of noise. Finally, computational efficiency and memory usage is discussed.

### 5.4.1 Classification Performance on the Diatom Set

The classification performance of our method on the Diatom set using the Max-tree algorithm is compared with the S.E. methods in Table 2. The length of the feature vectors for the methods is shown in the second column. Since the S.E. methods use normalized pattern spectra, the performance of our Max-tree method was tested for pattern spectra with and without normalization.

The results show that, on the original diatom set, the binned methods give the best results. The best two methods (S.E. BV binned and Max-tree) reach to a performance slightly

TABLE 3  
Classification Performances (%) on All Data Sets

Method	Diatoms	Brodatz	COIL-20	COIL-100
Max-tree	91.1 (1.6)	96.5 (0.6)	98.9 (0.5)	96.9 (0.6)
S.E. BV	93.8 (2.8)	82.9 (1.5)	99.0 (0.8)	97.4 (0.6)

Between brackets the corresponding standard deviations for the best two methods.

above 90 percent. When the Max-tree method is limited to use only size information, which is similar to the S.E. methods, the performance drops to 83.8 percent; the shape-only version of the Max-tree classified 83.0 percent correctly. It should be noted that there is no noticeable difference in computing time between these two Max-tree versions based on 1D pattern spectra and our proposed method based on 2D pattern spectra. The classification performance of several other methods were compared earlier [6]. It was then found that the best two performing ornamentation-based methods on the same diatom set were a combination of gray-level co-occurrence matrix (GLCM) and Gabor wavelets (84.3 percent) and a preliminary version of the Max-tree method described here (81.4 percent). Furthermore, Jalba et al. [15] reported another ornamentation-based method with a 88.1 percent classification performance on this set.

As Table 2 shows, normalization lowers the classification performance of the Max-tree methods on this set by roughly 10 percent. It reduces the sensitivity to difference in contrast and illumination. If contrast information contains meaningful information (as in the case of diatoms, where the contrast of certain features with respect to the background can differ between species), normalization leads to degraded performance. On the other hand, it also makes the method less sensitive to blurring (which tends to reduce contrast as well) caused, e.g., by artificial rotation.

#### 5.4.2 Classification Performance of Best Two Methods on All Sets

The classification performance and the corresponding standard deviation of S.E. BV and Max-tree binned for the four image sets used are shown in Table 3. As can be seen, the performance of these two methods is similar, except in the Brodatz case, where the Max-tree method significantly outperforms the S.E. BV method. In the other cases, the differences are not significant. We assume the reason for the different behavior of the methods on the Brodatz set is that these texture images often contain very long, but curly fiber-like structures which make it hard for methods that try to fit these “fibers” with S.E.s, whereas this does not pose a problem for connectivity-based methods. On the other hand, image details that are sometimes, but not always, connected (like some patterns in the diatom images), form a problem for the latter methods. This can be solved by using second-generation connectivities [2], [23], [31].

Recently, another texture-based method was described by Jalba et al. [15] that, on the Brodatz data set, achieved a 93.5 percent classification performance, which is high but still below that of the Max-tree method obtained here. The best out of six other methods compared in [15] has a performance of 93.9 percent on the Brodatz set, also below our method.

#### 5.4.3 Rotation-Invariance

In practical diatom analysis, diatom images are not aligned. Aligning every image manually to the same orientation (as was done for the diatom data set) is labor intensive. Although many diatom valve images could be aligned automatically, this causes problems for diatoms with a circular or triangular contour, but with nonsymmetric ornamentation patterns, like the one shown in Fig. 3a. Another problem is diatom shells as in Fig. 3b, where more than one orientation can be found in a single image. Preferably, a features used for classification should be invariant to rotation. In Fig. 4a, the performance of the methods is shown for images rotated over different angles only during the test phase. As can be seen, while the methods using the Max-tree algorithm with rotation-invariant attributes perform well for any rotation angle, the S.E. methods only perform well for the original image set, and due to the symmetry of most diatoms, also on 180 degree rotated images. S.E. Euclidean, which computes pattern spectra using a union of openings with five structuring elements, including four linear at 0, 45, 90, and 135 degrees, also performs well at those angles.

We rotated all diatom images digitally, which can introduce some blurring, possibly affecting the classification performance. This probably causes the residual rotation sensitivity seen in the plot for the nonnormalized Max-tree method. To estimate the significance of this effect, the methods were also applied by training on the original image set and using a test set whose images were blurred with a  $5 \times 5$  averaging kernel to simulate the blurring caused by rotation. Note that this  $5 \times 5$  kernel creates much more distortion than any single rotation. This blurring caused the following performance drops: the Max-tree method from 89.8 to 74.4 percent, S.E. BV from 92.4 to 74.6 percent, S.E. UV from 87.6 to 55.8 percent, and S.E. Euclidean from 75.8 to 24.1 percent. Thus, the influence of orientation on the performance for the Max-tree is probably caused by the blurring effect. This might also hold for the S.E. Euclidean, but not for the S.E. UV and the S.E. BV methods, since their performances are better with the heavily blurred test set than with the slightly blurred rotated test set.

To reduce the sensitivity of the S.E. methods to orientation, we used a training set of images rotated over angles: 0, 5, 10, 15, 30, 45, and 90 degrees. Fig. 4b shows that the S.E. methods perform well around the angles available in the training set, but their performances are low otherwise. The performance of the S.E. BV method for nonrotated images was reduced to 87 percent, i.e., below that of the Max-tree trained on the original set.

#### 5.4.4 Influence of Noise

The influence of noise on the classification performance of the methods was measured by evaluating image sets created from the original image set by adding Gaussian noise with standard deviations between  $\sigma = 0.01$  and 0.64. In Fig. 5a, the classification performance of the methods trained on the original and tested on the noisy image set is shown. We tried to reduce the effect of noise through various standard smoothing techniques, of which the blur  $5 \times 5$  averaging filter gave the best result. The results of using that preprocessing operator is shown in Fig. 5b. Notice in both columns that, although normalization slightly lowers the

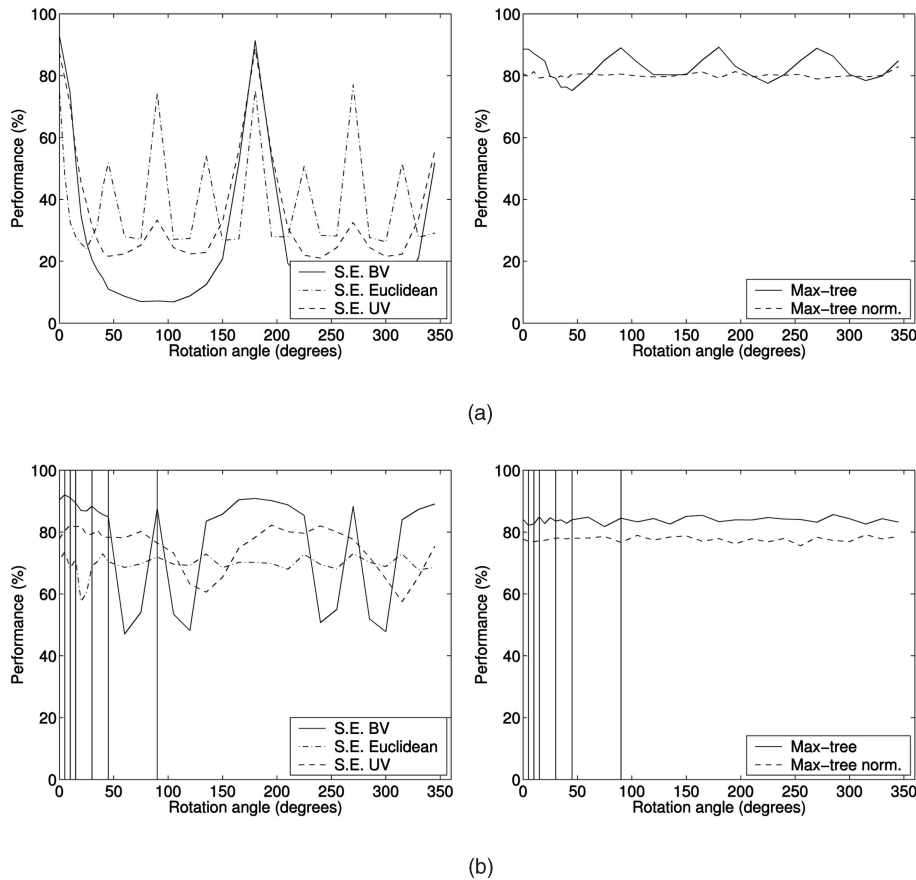


Fig. 4. Performance on rotated diatom images: (a) trained using aligned image set and (b) trained using rotated images, where the bars denote the angles used for training.

performance on (nearly) noise-free images, it clearly improves the performance for noisy images.

The performance of the methods on a noisy training and a noisy test set with the same  $\sigma$  is shown in Fig. 6. It can clearly be seen that the Max-tree methods are much more robust to noise than the S.E. methods. This can be explained by the fact that the noise affects mainly the low-area bins of the pattern spectrum in the Max-tree case, but it affects far more bins in the S.E. case. As an illustration, let image  $f$  contain a large white square. Placing black noise pixels in the square and white pixels outside it creates singleton sets to appear in the lowest area bin of the area pattern spectrum. Unless the binning of the spectrum is very fine, the large square should stay in the same bin of the area spectrum. By contrast, in the case of pattern spectra computed using a linear, horizontal S.E., if a black noise pixel appears at some fraction  $\epsilon$  of the width  $w$  of the square, the signal from the bin corresponding to  $w$  will be moved to bins corresponding to  $(1 - \epsilon)w$  and  $\epsilon w$ . Given spatially uniformly distributed noise pixels, this will lead to a uniform distribution of signal over bins below that corresponding to  $w$ , apart from the additional noise at the smallest widths caused by white pixels in the background.

#### 5.4.5 Computation Time

The average computation time of the binned methods (with optimal parameters and compiled with the same compiler

settings) on a 2.8 GHz Pentium4 PC is shown in Fig. 7 for two image sets: the ADIAC set of diatom images and the Brodatz set. As can be seen, the two methods with the highest classification performance (S.E. BV and Max-tree binned) are also the fastest. Furthermore, it is clear that, for the Max-tree methods, the time needed to compute a feature vector is independent of the size of the pattern spectrum and, thus, of the feature vector length, as opposed to the S.E. methods. The average computation time per pixel for the Max-tree and S.E. BV method with optimal feature vector length on the diatom set was  $0.8 \mu\text{s}$  and  $7.1 \mu\text{s}$ , respectively. For the Brodatz set this was, respectively,  $1.3 \mu\text{s}$  and  $6.5 \mu\text{s}$ . The computation time of the S.E.-methods is independent of image content, which is not the case for the Max-tree methods.

#### 5.4.6 Memory Usage

For an input image containing  $N$  pixels, our current Max-tree implementation uses a pixel queue and a status array, both containing  $N$  elements. Furthermore, for every peak component, a node is used that contains the corresponding parent pointer, intensity level, and the attribute values  $A$  and  $I/A^2$ . Besides the input image,  $O(N)$  integers and  $O(N)$  floats are used. The S.E. methods use an output image. For the S.E. UV and the S.E. Euclidean methods,  $O(M + W + H)$  integers are needed for the dilation and erosion algorithm, where  $M$  is the maximum of the width  $W$  and the height  $H$  of the image. Furthermore, in the implementation of these methods, an

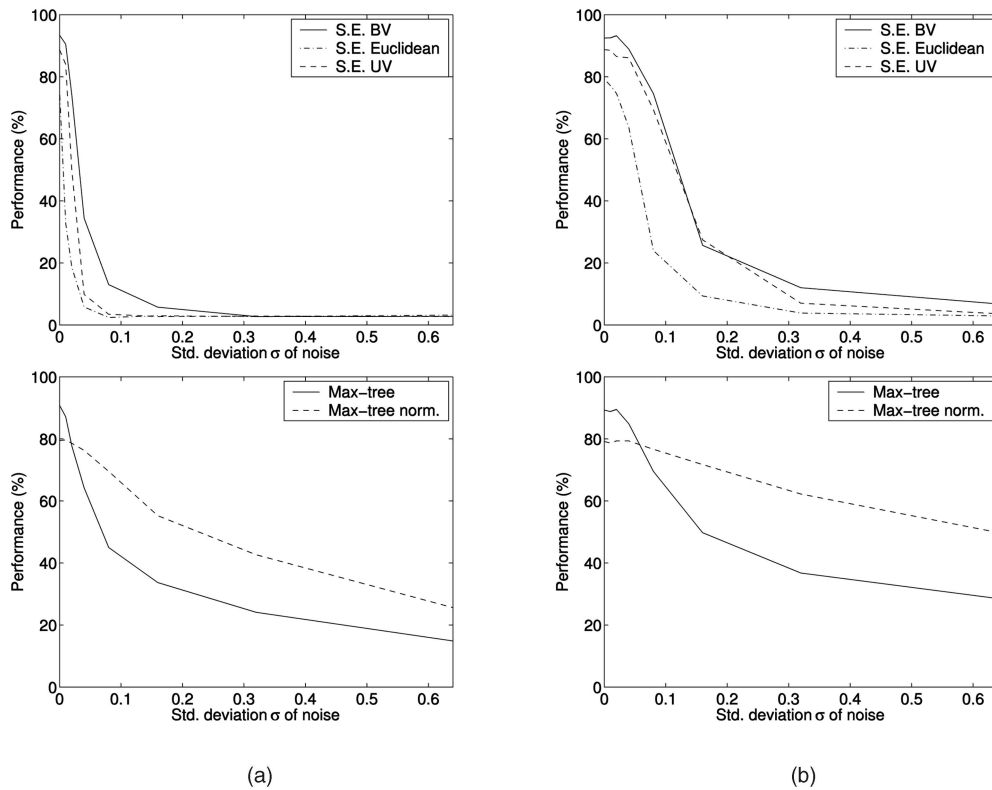


Fig. 5. Performance on noisy test set using the original training diatom set. (a) Noisy test set and (b) noisy test set blurred with a  $5 \times 5$  averaging filter.

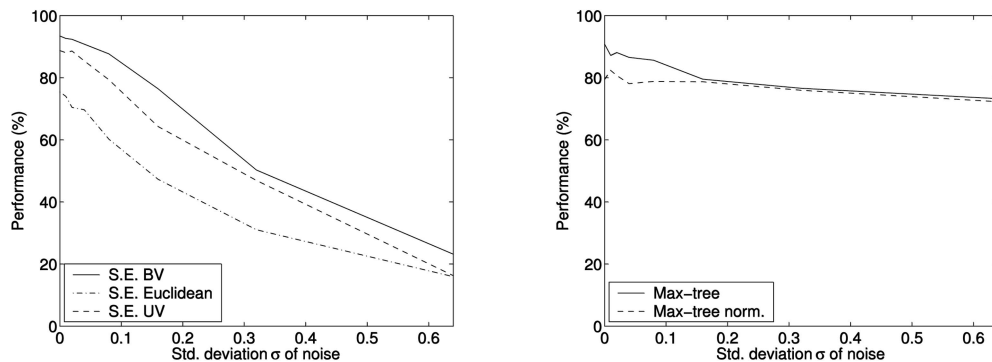


Fig. 6. Performance on noisy diatom training and test sets with identical noise level.

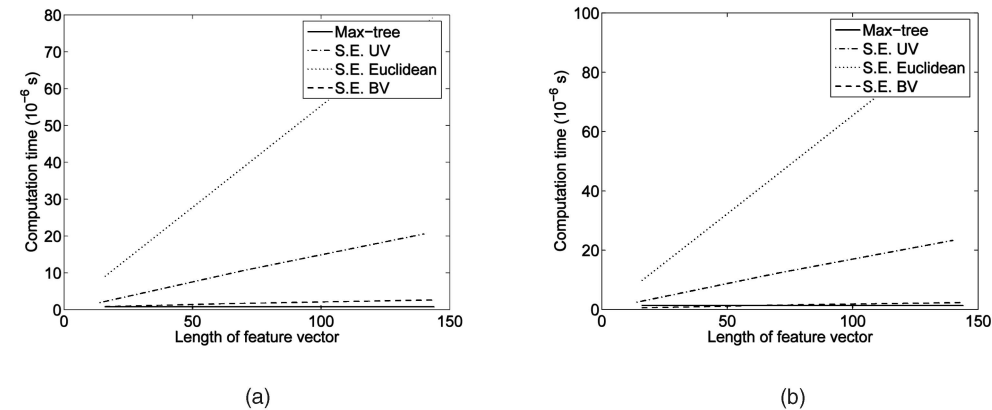


Fig. 7. Computation time per pixel versus length of feature vector for the binned methods. (a) Diatoms. (b) Brodatz.

auxiliary image is used to save some computations. The S.E. BV method uses  $O(M)$  integers and, for faster computation,  $w + h$  auxiliary images for a  $w \times h$  pattern spectrum. To

avoid clipping problems with dilations and erosions using nonflat S.E., the S.E. methods use 32-bit signed integer gray-scale images instead of the 8-bit gray-scale images used by the

Max-tree method. Thus, the S.E. UV and the S.E. Euclidean generally use the least memory. Memory-efficient implementation of S.E. BV leads to a strong increase in computing time.

## 6 DISCUSSION AND CONCLUSIONS

Pattern spectra for the computation of feature vectors are very suitable for image classification. In this paper, we introduced a multiscale and multishape morphological method for pattern-based analysis and classification of gray-scale images using connected operators, and an efficient implementation using Salembier's Max-tree algorithm. Our method performs a global analysis of the patterns in an image. This type of information can be of use in many other applications, such as content-based image retrieval, soil analysis [33], or the classification of pollen grains [26], where all grains are more or less round, but whose patterns are not always symmetric.

We compared our new method with existing methods using structuring elements (S.E. methods). Methods using moments of pattern spectra were outperformed by methods using binned pattern spectra. This may be due to the sensitivity of moments (in particular, with high order) to outliers in the distribution. For image sets where the images are symmetrical or manually aligned to have the same orientation, our method was never significantly worse than the best S.E. competitor, and, in one case, significantly better (Brodatz). When rotation-invariance is desired, like in the case of images with different orientations, our method with its rotation-invariant attributes is preferable over methods based on linear structuring elements, since the latter are not invariant to rotation. Although rotation-invariance can be approximated by using linear structuring elements at many orientations, this leads to a sharp increase in computation time, which is not desirable when large sets of images are used. A further advantage of our method is the separation of size and shape information, which means that a class of image components with similar shape, but varying size, can now be treated as one class for filtering or analysis. Furthermore, when pattern spectra with a large number of size or shape classes need to be computed, our method is also preferable since methods based on structuring elements need to filter an image for every class used, while the time needed to compute a pattern spectrum using our method is independent of the size of that pattern spectrum. In the current setting, a speed gain of between 5 and 9-fold per image set was obtained. Also, it was shown that our method is significantly less sensitive to noise than any of the compared S.E. methods.

In future work, we will investigate other attributes such as moment invariants of Hu [12] or the affine moment invariants of Flusser and Suk [7], which may improve the performance of our method further. For texture segmentation of images, local texture measures are necessary, and adaptations of our method are needed.

## APPENDIX

The following list summarizes the notational conventions used in this paper:

- $E$ : image domain ( $\mathbb{R}^n$  or  $\mathbb{Z}^n$ ).
- $\mathcal{P}(E)$ : power set (set of all subsets) of  $E$ .
- $X$ : binary image.
- $f$ : gray-scale image.

- $X^h$ : translate of set  $X \subseteq E$  by vector  $h$ .
- $X^c$ : complement of set  $X \subseteq E$ .
- $tX$ : scaling of  $X$  by factor  $t \in \mathbb{R}$ .
- $tf$ : scaling of  $f$  by factor  $t \in \mathbb{R}$ .
- $B_r^x$ : ball of radius  $r$  and center  $x$ .
- S.E.: structuring element
- $X \oplus B$ : dilation of  $X$  by  $B$ .
- $X \ominus B$ : erosion of  $X$  by  $B$ .
- $X \circ B$ : opening of  $X$  by  $B$ .
- $X \bullet B$ : closing of  $X$  by  $B$ .
- $X \otimes B$ : hit-or-miss transform of  $X$  by  $B = (B_1, B_2)$ .
- $X \oslash B$ : thinning of  $X$  by  $B = (B_1, B_2)$ .
- $A(X)$ : Lebesgue measure (area, volume) of  $X$ .
- $A(f)$ : integral (sum of gray levels) of  $f$ .
- $\Lambda$ : totally ordered set (e.g.,  $\mathbb{R}$  or  $\mathbb{Z}$ ).
- $\mathcal{V}_h(f)$ : level set of  $f : \{x \in E | f(x) = h\}$ .
- $\mathcal{T}_h(f)$ : threshold set of  $f : \{x \in E | f(x) \geq h\}$ .
- flat zone connected component of  $\mathcal{V}_h(f)$
- $I_h^f$ : index set of flat zones of  $f$  at level  $h$ .
- $L_h^k(f)$   $k$ th: flat zone of  $f$  at level  $h$ .
- $P_h^k(f)$   $k$ th: peak component (connected component of  $\mathcal{T}_h(f)$ ).
- $N_h^k(f)$ : Max-tree node corresponding to  $P_h^k(f)$  (subset of  $P_h^k$  with gray level  $h$ ).
- grain (of a binary image  $X$ ): connected component of  $X$ .
- grain (of a gray-scale image  $f$ ): peak component of  $f$ .
- $C_i^X$   $i$ th: grain of  $X$ .
- $C_i^{\mathcal{T}_h(f)}$   $i$ th: grain of  $\mathcal{T}_h(f)$  (i.e.,  $P_h^i(f)$ ).
- $\{\Gamma_r\}_{r \in \Lambda}$ : binary size granulometry.
- $\{\gamma_r\}_{r \in \Lambda}$  gray-scale size granulometry.
- $s_\Gamma(X)$ : pattern spectrum of  $X$  corresponding to binary size granulometry  $\{\Gamma_r\}$ .
- $s_\gamma(f)$ : pattern spectrum of  $f$  corresponding to gray-scale size granulometry  $\{\gamma_r\}$ .
- $\Gamma_x(X)$ : connected opening of binary image  $X$  at point  $x \in E$  (Section 2.3).
- $T(C)$ : criterion  $T$  applied to grain  $C$ .
- $\Phi_T(C)$ : trivial thinning of grain  $C$  by criterion  $T$ .
- $\Phi^T(X)$ : binary attribute thinning of  $X$  by criterion  $T$ .
- $\tau(P_h^k)$ : attribute value of  $P_h^k$ .
- $\{\Phi_r\}_{r \in \Lambda}$ : binary shape granulometry.
- $\{\phi_r\}_{r \in \Lambda}$ : gray-scale shape granulometry.
- $\{\Phi_r^\tau\}_{r \in \Lambda}$ : binary shape granulometry with criterion  $T(C) = (\tau(C) \geq r)$ .
- $\{\phi_r^\tau\}_{r \in \Lambda}$ : gray-scale shape granulometry with criterion  $T(P_h^k) = (\tau(P_h^k) \geq r)$ .

## REFERENCES

- [1] S. Batman and E.R. Dougherty, "Size Distributions for Multivariate Morphological Granulometries: Texture Classification and Statistical Properties," *Optical Eng.*, vol. 36, pp. 1518-1529, 1997.
- [2] U. Braga-Neto and J. Goutsias, "A Theoretical Tour of Connectivity in Image Processing and Analysis," *J. Math. Imaging and Vision*, vol. 19, pp. 5-31, 2003.
- [3] E.J. Breen and R. Jones, "Attribute Openings, Thinnings and Granulometries," *Computer Vision and Image Understanding*, vol. 64, pp. 377-389, 1996.
- [4] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.
- [5] Y. Chen and E.R. Dougherty, "Gray-Scale Morphological Granulometric Texture Classification," *Optical Eng.*, vol. 33, pp. 2713-2722, 1994.

- [6] *Automatic Diatom Identification*, J.M.H. Du Buf and M.M. Bayer, eds., series on Machine Perception and Artificial Intelligence, World Scientific Publishing, 2002.
- [7] J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants," *Pattern Recognition*, vol. 26, pp. 167-174, 1993.
- [8] P. Ghosh and B. Chanda, "Bi-Variate Pattern Spectrum," *Proc. IEEE CS Brazilian Symp. Computer Graphics and Image Processing (SIBGRAPI '98)*, pp. 476-483, Oct. 1998.
- [9] J. Gil and R. Kimmel, "Efficient Dilation, Erosion, Opening, and Closing Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1606-1617, 2002.
- [10] R. Gonzalez and R. Woods, *Digital Image Processing*, second ed. Addison-Wesley, 2002.
- [11] H.J.A.M. Heijmans, "Connected Morphological Operators for Binary Images," *Computer Vision and Image Understanding*, vol. 73, pp. 99-120, 1999.
- [12] M.K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Information Theory*, vol. 8, pp. 179-187, 1962.
- [13] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153-158, 1997.
- [14] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink, "Automatic Segmentation of Diatom Images," *Microscopy Research and Technique*, vol. 65, pp. 72-85, 2004.
- [15] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink, "Morphological Hat-Transform Scale Spaces and Their Use in Pattern Classification," *Pattern Recognition*, vol. 37, pp. 901-915, 2004.
- [16] R. Jones, "Connected Filtering and Segmentation Using Component Trees," *Computer Vision and Image Understanding*, vol. 75, pp. 215-228, 1999.
- [17] P. Maragos, "Pattern Spectrum and Multiscale Shape Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 701-715, 1989.
- [18] G. Matheron, *Random Sets and Integral Geometry*. John Wiley, 1975.
- [19] A. Meijster and M.H.F. Wilkinson, "Fast Computation of Morphological Area Pattern Spectra," *Proc. Int'l Conf. Image Processing*, pp. 668-671, 2001.
- [20] A. Meijster and M.H.F. Wilkinson, "A Comparison of Algorithms for Connected Set Openings and Closings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 484-494, 2002.
- [21] P.F.M. Nacken, "Chamfer Metrics in Mathematical Morphology," *J. Math. Imaging and Vision*, vol. 4, pp. 233-253, 1994.
- [22] P.F.M. Nacken, "Chamfer Metrics: The Medial Axis and Mathematical Morphology," *J. Math. Imaging and Vision*, vol. 6, pp. 235-248, 1996.
- [23] G.K. Ouzounis and M.H.F. Wilkinson, "Second-Order Connected Attribute Filters Using Max-Trees," *Proc. Int'l Symp. Math. Morphology: 40 Years on*, pp. 65-74, Apr. 18-20, 2005.
- [24] R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [25] I. Ragnemalm, "Fast Erosion and Dilation by Contour Processing and Thresholding of Distance Maps," *Pattern Recognition Letters*, vol. 13, pp. 161-166, 1992.
- [26] O. Ronneberger, H. Burkhardt, and E. Schultz, "General-Purpose Object Recognition in 3D Volume Data Sets Using Gray-Scale Invariants—Classification of Airborne Pollen-Grains Recorded with a Confocal Laser Scanning Microscope," *Proc. 16th Int'l Conf. Pattern Recognition*, vol. 2, pp. 290-295, Sept. 2002.
- [27] P. Salembier, A. Oliveras, and L. Garrido, "Anti-Extensive Connected Operators for Image and Sequence Processing," *IEEE Trans. Image Processing*, vol. 7, pp. 555-570, 1998.
- [28] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," *IEEE Trans. Image Processing*, vol. 4, pp. 1153-1160, 1995.
- [29] F. Sand and E.R. Dougherty, "Statistics of the Morphological Pattern Spectrum Moments for a Random Grain Model," *J. Math. Imaging and Vision*, vol. 1, pp. 121-135, 1992.
- [30] J. Serra, *Image Analysis and Mathematical Morphology*, vol. 1, second ed. Academic Press, 1982.
- [31] J. Serra, "Connectivity on Complete Lattices," *J. Math. Imaging and Vision*, vol. 9, pp. 231-251, 1998.
- [32] K. Sivakumar, M.J. Patel, N. Kehtarnavaz, Y. Balagurunathan, and E.R. Dougherty, "A Constant-Time Algorithm for Erosions/Dilations with Applications to Morphological Texture Feature Computation," *Real-Time Imaging*, vol. 6, pp. 223-239, 2000.
- [33] A. Sofou, C. Tzafestas, and P. Maragos, "Segmentation of Soilsection Images Using Connected Operators," *Proc. Int'l Conf. Image*, pp. 1087-1090, 2001.
- [34] P. Soille, E. Breen, and R. Jones, "Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 562-567, 1996.
- [35] P. Soille and H. Talbot, "Directional Morphological Filtering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1313-1329, 2001.
- [36] E.F. Stoermer and J.P. Smol, *The Diatoms: Applications for the Environmental and Earth Sciences*. Cambridge Univ. Press, 1999.
- [37] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Connected Rotation-Invariant Size-Shape Granulometries," *Proc. 17th Int'l Conf. Pattern Recognition*, vol. 1, pp. 688-691, Aug. 2004.
- [38] E.R. Urbach and M.H.F. Wilkinson, "Shape-Only Granulometries and Grey-Scale Shape Filters," *Proc. Int'l Symp. Math. Morphology*, pp. 305-314, 2002.
- [39] M. van Herk, "A Fast Algorithm for Local Minimum and Maximum Filters on Rectangular and Octagonal Kernels," *Pattern Recognition Letters*, vol. 13, pp. 517-521, 1992.
- [40] L. Vincent, "Granulometries and Opening Trees," *Fundamenta Informaticae*, vol. 41, pp. 57-90, 2000.
- [41] M.H.F. Wilkinson and M.A. Westenberg, "Shape Preserving Filament Enhancement Filtering," *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, W.J. Niessen and M.A. Viergever, eds., pp. 770-777, 2001.



IEEE and the IEEE Computer Society.



image processing and tomographic reconstruction. He was appointed associate professor in 1992 and full professor in 2003, respectively, at the Institute for Mathematics and Computing Science of the University of Groningen, where he currently holds a chair in scientific visualization and computer graphics. His current research interests include biomedical visualization, neuroimaging, and bioinformatics. He is a senior member of the IEEE.



working on simulating the intestinal microbial ecosystem on parallel computers. During that time, he edited the book *Digital Image Analysis of Microbes* (John Wiley, 1998) together with Frits Schut. After this, he worked as a researcher at the IWI on image analysis of diatoms. He is currently an assistant professor at the IWI. He is a senior member of the IEEE and the IEEE Computer Society.

**Erik R. Urbach** received the MSc degree in computer science from the Institute of Mathematics and Computing Science, University of Groningen (RUG) in 2002, where he now works on the Connected Morphological Operators for Scale and Shape Spaces (C-MOSSS) project toward the PhD degree. The prime areas of his research are connected filters, multiscale and multishape analysis, image classification, and texture analysis. He is a student member of the

**Jos B.T.M. Roerdink** received the MSc degree (1979) in theoretical physics from the University of Nijmegen, the Netherlands. After receiving the PhD degree (1983) from the University of Utrecht and a two-year position (1983-1985) as a postdoctoral fellow at the University of California, San Diego, both in the area of stochastic processes, he joined the Centre for Mathematics and Computer Science in Amsterdam. There, he worked from 1986-1992 on

**Michael H.F. Wilkinson** received the MSc degree in astronomy from the Kapteyn Laboratory, University of Groningen (RUG) in 1993, after which he worked on image analysis of intestinal bacteria at the Department of Medical Microbiology, RUG. This work formed the basis of his PhD at the Institute of Mathematics and Computing Science (IWI), RUG, in 1995. He was appointed as a researcher at the Centre for High Performance Computing (also RUG) working on simulating the intestinal microbial ecosystem on parallel computers. During that time, he edited the book *Digital Image Analysis of Microbes* (John Wiley, 1998) together with Frits Schut. After this, he worked as a researcher at the IWI on image analysis of diatoms. He is currently an assistant professor at the IWI. He is a senior member of the IEEE and the IEEE Computer Society.