

# Pattern Recognition Using Morphological Class Distribution Functions and Classification Trees

Marcin Iwanowski and Michal Swiercz

Institute of Control and Industrial Electronics,  
Warsaw University of Technology,  
ul.Koszykowa 75, 00-662 Warszawa Poland  
{iwanowski,swierczm}@ee.pw.edu.pl

**Abstract.** The paper presents an effective and robust method of classifying binary patterns. It starts with classification of foreground pixels of binary image into several spatial classes, which is performed using morphological image processing. By performing this classification with structuring elements of increasing sizes, the spatial class distribution functions are produced. These functions are normalized and sampled in order to obtain feature vectors of constant length that are invariant to pattern translation, rotation and scaling. Such feature vectors are next used to perform tree-based classification.

## 1 Introduction

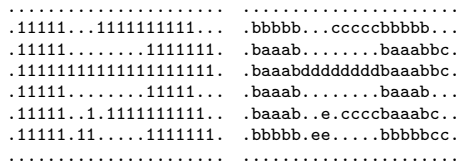
In this paper, a method for recognizing binary patterns using morphological class distribution functions and decision trees is presented. The method is based on morphological classification. It allows extracting from the binary image pixels belonging to different *spatial classes* consisting of pixels characterized by particular morphological properties. Depending on the class being detected, various class extractors can be defined, based on morphological image processing operations. All the operators leading to extraction of spatial classes are using a single parameter – the structuring element of morphological operators. By applying structuring elements of increasing size when extracting spatial classes, *class distribution functions* can be obtained. They are expressing the dependence of the number or ratio of pixels belonging to a given spatial class on the size of the structuring element. The shape and characteristics of class distribution functions depend on the pattern for which they are computed. In the method presented in this paper, this function is normalized and sampled into a given number of samples using cubic spline interpolation. Thanks to this procedure, the scale-invariant feature vector of constant length is obtained. This trait, along with translation and (under some conditions) rotation invariance, make these features an effective tool for pattern recognition. In the proposed method, they are used as the input for tree-based classification. Classification trees, apart from their principal task – classification – allow also finding features that have a real influence on the classification result. An additional pre-processing of the training set is proposed in this paper – the feature preselection. By analysing the

scattering measure for all pattern classes, features that are characterized by low in-class integrity are removed from this set. Training set with preselected features is finally used to train the classification tree. This tree is tested using a separate testing set.

The paper consists of 6 sections. In section 2, the idea and extraction method of morphological spatial classes are described. Section 3 presents the class distribution functions and their sampling. Section 4 describes the decision tree classification. In section 5, the test results are presented and finally, section 6 concludes the paper.

## 2 Morphological Spatial Classes

The classification task aims at assigning elements of the feature space into appropriate classes consisting of elements similar one to another. The the first application of mathematical morphology [6,7,9] to binary pattern classification was described in [5]. It was focused on forest analysis based on forest masks computed from remotely sensed images. Mathematical morphology tools were used to classify forest regions into 7 spatial classes. The applications of this methodology (called MSPA – morphological spatial patterns analysis) into forest pattern detection was also described in [3]. In [1,2,4] more generic view on this methodology was presented, allowing it to be applied to classify regions of various binary patterns: electronic circuit boards, water masks and binary shapes of various kinds.



**Fig. 1.** Binary pattern (left) and five classes (right): a - core, b – core boundary, c – branches, d – corridors, e – isolated

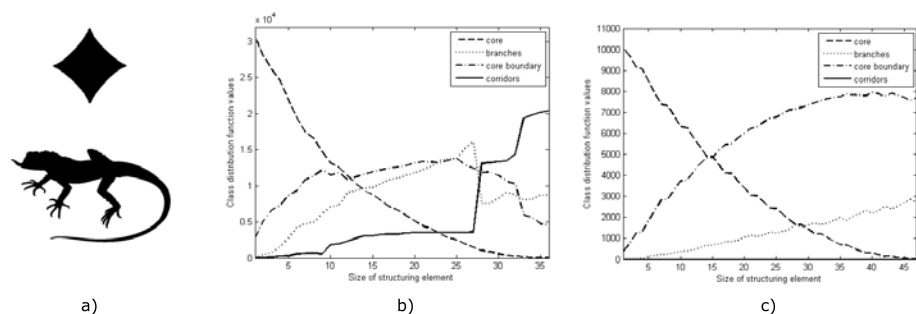
The *morphological spatial class* of the binary image  $F$  is defined as a subset of foreground pixels characterized by a particular spatial property. Depending on the particular spatial class characteristics, various morphological operators should be applied to extract it. The only parameter used in the class extraction process is a structuring element  $B$ . In this paper we define the following set of spatial classes:

1. *Core* – region consisting of foreground pixels that are farther from the boundary of  $F$  than a distance implied by  $B$ . This class is obtained by means of erosion operator:  $\Psi_{cr}(F, B) = F \ominus B$ .

2. *Isolated* – connected components of the input image that do not contain any core pixel. This class is the residue of the morphological reconstruction of input image with core regions used as markers:  $\Psi_{is}(F, B) = F \setminus F \Delta \Psi_{cr}(F, B)$ .
3. *Core boundary* – region of pixels that are located inside the initial object that do not belong to the core region and are not farther from the core than the distance implied by  $B$ . This class can be obtained as a difference between opening and erosion:  $\Psi_{cb}(F, B) = (F \circ B) \setminus (F \ominus B) = (F \circ B) \setminus \Psi_{cr}(F, B)$ .
4. *Corridors* – groups of pixels which are neither core nor core boundary and which connect two disjoint core regions. A single object (connected component of foreground pixels) can contain more than just one core region. A connector between all cores of a single object that does not belong to core boundary is a corridor. Contrary to core boundary pixels, the corridor pixels are in a distance from the cores greater than implied by  $B$ . This class can be obtained by means of anchored homotopic skeletonization [6] of the input image with core pixels considered as anchor ones:  $\Psi_{co}(F, B) = SKH(F, \Psi_{cb}(F, B))$ , where  $SKH(F, G)$  stands for the anchored homotopic skeleton of  $F$  with anchor pixels  $G$ .
5. *Branches* – groups of pixels which are neither core nor core boundary but are attached to a single core region (dead-ends of pattern):  $\Psi_{br}(F, B) = F \setminus (\Psi_{is}(F, B) \cup \Psi_{cr}(F, B) \cup \Psi_{cb}(F, B) \cup \Psi_{co}(F, B))$ .

The example of classification into five above classes for  $B$  equal to elementary 8-connected structuring element is shown on Fig. 2.

The result of classification strongly depends on the applied structuring element  $B$  – on its shape and size. These parameters imply the form of the pixel neighborhood considered. Consequently, they imply also the distance from a central pixel of the structuring element to other pixels belonging to it. Depending on the type of the structuring element various distance measures are considered. The elementary structuring elements induces either city-block distance (4-connected element) or max-norm distance (8-connected). Assuming that the distance from



**Fig. 2.** Class distribution functions of *core*, *core boundary*, *branch* and *corridor* for test patterns: “lizard” (b) and “diamond” (c). Patterns are shown in (a).

pixels belonging to  $B$  is not greater than one, as in the above case, the notation  $B^{(n)}$  will refer to a larger structuring element, which contains pixels not farther than  $n$  from the central pixel.  $B^{(n)}$  is thus the neighborhood of radius  $n$ . The structuring element  $B^{(n)}$  can be defined in various ways. The simplest (and fastest) is based on superposition by successive dilations of  $n$  elementary structuring elements  $B^{(n)} = B \oplus B \oplus \dots \oplus B$ , where  $B$  stands for an elementary structuring element. In order to get the neighborhood of radius  $n$  according to the *Euclidean* distance the superposition by dilations cannot be used and the structuring element  $B^{(n)}$  have to be computed individually for every  $n$ . Another possible choice of  $B^{(n)}$  is the octagon-shaped element, that can be obtained by alternate usage of 4- and 8-connected elementary ones.

### 3 Class Distribution Functions and Feature Extraction

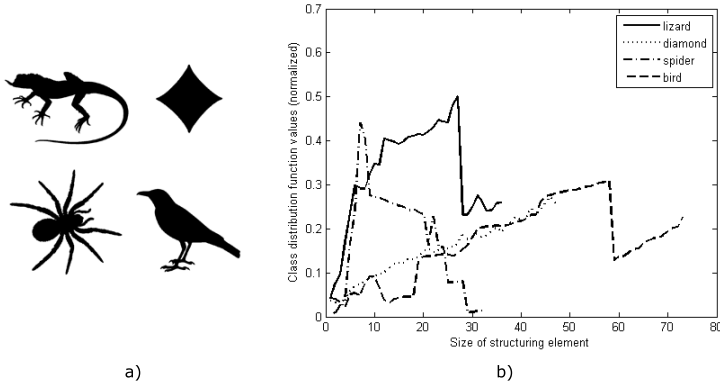
The results of spatial pixel classification depends on the size  $n$  of the structuring element  $B^{(n)}$ . Moreover this dependence differs from one binary pattern to another. The classification of image pixels using the series of structuring elements  $B^{(n)}$  for increasing  $n$  allows obtaining the *class distribution functions* for each class. They are defined as:

$$\mathcal{D}_{CL}(n) = |\Psi_{CL}(F, B^{(n)})|, \quad (1)$$

where  $|\cdot|$  stands for the number of pixels of the argument and  $CL \in \{cr, is, cb, co, br\}$  refers to the spatial class. Class distribution functions of four spatial classes (class *isolated* is not applicable in this case) of a test binary input pattern is presented in Fig. 2. It shows that functions of different shapes are noticeably different from one another.

The class distribution functions described above can be treated as an extension of granulometry. Granulometry by opening is equal to  $\mathcal{D}_{cr}(n) + \mathcal{D}_{cb}(n)$ , by opening by reconstruction to  $\mathcal{D}_{cr}(n) + \mathcal{D}_{cb}(n) + \mathcal{D}_{co}(n) + \mathcal{D}_{br}(n)$ .

The class distribution function defined by the Eq. 1 has some important properties. First, it depends on the form and size of the object(s) for which it is computed (see Fig. 2). This property makes it suitable for recognizing binary patterns. Second, it is always invariant to translations of objects within the image – this comes directly from the obvious property of morphological operators. Third, to some extent it is invariant to rotation. The extent depends on the type of the structuring element  $B^{(n)}$ . For elementary 4- and 8-connected elements it is invariant to rotations by  $\frac{\pi}{2}$ , in case of octagonal element – by  $\frac{\pi}{4}$ . In case of an Euclidean disk – by any angle. Finally, the class distribution function is not scale-invariant: scaling of the binary pattern implies scaling of the function defined by Eq. 1. This can be, however, countered by the normalization and sampling technique described further in this paper.



**Fig. 3.** Class distribution functions of class *branch* (b) for different test patterns (a)

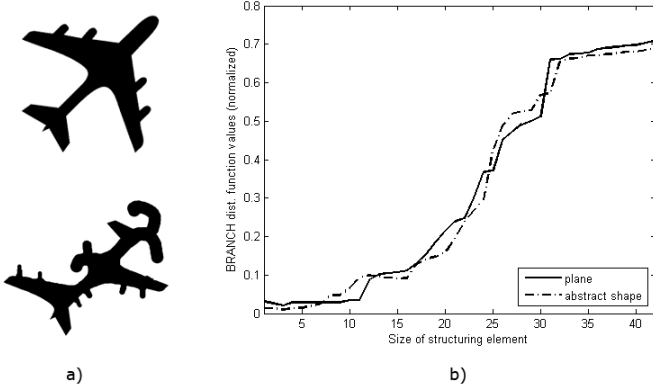
In order to get scale-invariance, at first, the normalization along “y” axis is applied. It is obtained by dividing values of the function by a normalization factor equal to the total number of foreground pixels of the input image  $F$ :

$$D'_{CL}(n) = \frac{|\Psi_{CL}(F, B^{(n)})|}{|F|} \tag{2}$$

In Fig. 3, normalized distribution functions of the class *branch* for various patterns are presented. It is worth noting at this point, that since class distribution functions depend only on the number of pixels of a certain class but not in any way on the position of these pixels within the image, it is possible to propose visually different shapes having highly similar class distribution functions (an example case is presented in Fig. 4). This very rare case, however, did not present itself during tests, in which shapes derived from physical objects were used.

Pattern scale influences also another parameter – the range of sizes of the structuring elements (“x” axis of the distribution function). The maximum effective value of  $n$  in Eq. 2 for which this function may change equals  $n_{MAX}$ , which is the largest size of erosion such that  $F$  does not disappear completely<sup>1</sup>. For all  $n > n_{MAX}$  there are no more pixels belonging to the *core* class and all pixels are classified as *isolated*. For all  $n \leq n_{MAX}$ , at least two classes for each argument of the class distribution function always exist. Value  $n_{MAX}$  depends on the scale of a binary pattern. Assuming, for example, a binary pattern enlarged twice, one can observe that – comparing distribution functions of the original and the enlarged pattern – the latter has the same form as the former, but is stretched along the “x” axis and  $n_{MAX}$  is multiplied by 2. The values of a distribution function cannot be thus considered as scale-invariant features of patterns. Also, from the pattern recognition point of view, the feature vector

<sup>1</sup> In case of  $F$  foreground consisting of a single connected  $F$  component, this is the size of erosion that produces ultimately eroded set.



**Fig. 4.** Different shapes having a similar class distribution function: plane and abstract shape (a), and their “branch” class distribution functions (b)

consisting of features taken from the class distribution functions should have the same length independently from the range of structuring element sizes. This condition is however not fulfilled in the current case.

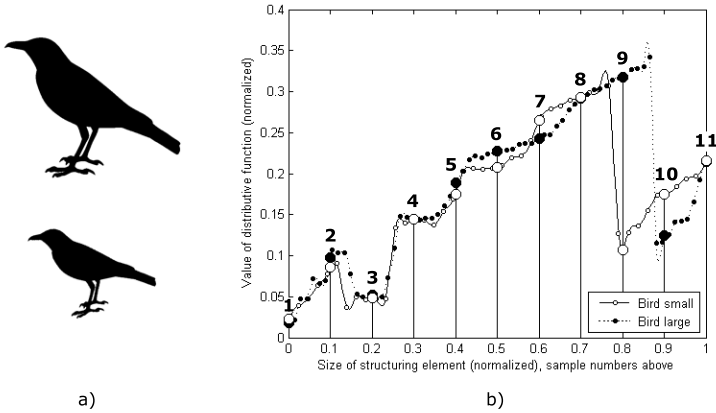
In order to produce a feature vector of the constant length, the class distribution function is sampled into a given number of samples  $s_{CL}$ . Due to the fact that sampling may require the values of the distribution function for real arguments, some interpolation is required. In the sampling method used in the experiments, the cubic spline interpolation was applied. As a result of sampling, the function defined by Eq. 2 of variable length is reduced to a given number of samples  $s_{CL}$ . The samples of distribution functions will be denoted using upper index in brackets and put together into a feature vector of the class:  $\mathbf{v}_{CL} = [\mathcal{D}_{CL}^{(1)}, \mathcal{D}_{CL}^{(2)}, \dots, \mathcal{D}_{CL}^{(s_{CL})}]$ . An example of sampling is presented in Fig. 5. Two normalized class distribution functions of variants of the same pattern (“bird”) are shown in a way that “x” axis was also normalized as:  $n' = \frac{n}{n_{MAX}}$ .

In order to get a complete morphological signature of the pattern, the feature vectors of all classes are grouped in a single feature vector of length  $s = s_{cr} + s_{is} + s_{cb} + s_{co} + s_{br}$ :

$$\mathbf{v} = [v^{(1)}, v^{(2)}, \dots, v^{(s)}] = [\mathbf{v}_{cr}, \mathbf{v}_{is}, \mathbf{v}_{bc}, \mathbf{v}_{co}, \mathbf{v}_{br}], \tag{3}$$

where  $v^{(i)}$  stands for the  $i$ -th element of the feature vector ( $i$ -th feature, always  $v^{(i)} \equiv \mathcal{D}_{CL}^{(j)}$  for certain  $j$  and  $CL$ ). In case of patterns consisting of a single connected component, there is no need to use class *isolated* since it would always be empty. In such case, the feature vector of length  $s = s_{cr} + s_{cb} + s_{co} + s_{br}$  is equal to:

$$\mathbf{v} = [v^{(1)}, v^{(2)}, \dots, v^{(s)}] = [\mathbf{v}_{cr}, \mathbf{v}_{bc}, \mathbf{v}_{co}, \mathbf{v}_{br}]. \tag{4}$$



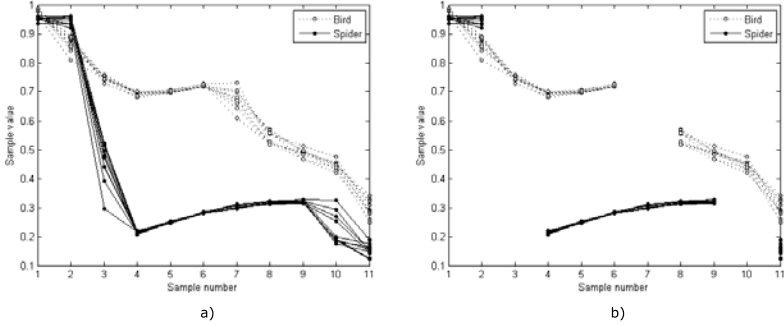
**Fig. 5.** Sampling of the *branch* class distribution functions (b) of two different variants of the “bird” pattern (a)

### 4 Tree-Based Classification

Classification (decision) trees [8] are graph structures where each node represents a certain decision rule, involving a test based on values of one of more features of the data set. These progressive tests divide the original dataset into disjoint subset nodes with higher class uniformity than the parent node. The final subsets which are not exposed to further divisions are called leaves and determine the class association of a case belonging to such a node. The number of leaves determines the tree size, while the number of edges between the root and the most distant leaves informs about the tree depth.

In order to perform the classification task, tree growing process is needed. It involves choosing the test conditions for each node, basing on a chosen quality criterion, in order to achieve the highest possible pattern class<sup>2</sup> uniformity in child nodes. The essential component of the tree-growing process is a training data set, i.e. a dataset, consisting of feature vectors of all pattern classes to be recognized. Moreover, each pattern type should be represented by multiple feature vectors computed for various pattern variations (also scaled, rotated, with disturbed boundary etc.). Let  $k$  be the number of all pattern classes. The set of feature vectors of all patterns in  $i$ -th pattern class is denoted as  $V_i$ . In fact this will be a matrix such that columns refer to features, while rows – to particular patterns of  $i$ -th type. In other words:  $V_i = [\mathbf{v}_{i,1}^T, \mathbf{v}_{i,2}^T, \dots, \mathbf{v}_{i,p_i}^T]^T$ , where  $\mathbf{v}_{i,j}$  stands for the feature vector (Eq. 3) of  $j$ -th example of  $i$ -th pattern type,  $p_i$  is the total number of examples of  $i$ -th pattern type in the training set and

<sup>2</sup> The notion of class is used in this paper in two meanings. *Spatial* class refer to the set of foreground pixles of the pattern (e.g. *core*, *branch*, etc.), while *pattern* class – to type of the pattern (e.g. “bird”, “spider”, etc.).



**Fig. 6.** Sampled features, class *core boundary* ( $\mathcal{D}_{cb}^{(1)}, \dots, \mathcal{D}_{cb}^{(11)}$ ) – without feature preselection (a), with feature preselection (b)

the upper index  $T$  stands for vector transposition. The whole training set is thus defined as  $V = \{V_1, V_2, \dots, V_k\}$ .

As it was pointed out earlier, normalized and sampled class distribution functions, in general, are fairly invariant to rotation and scaling. However, under certain conditions (for example, when using a structuring element that is highly asymmetrical in respect to rotations), some features can show a level of undesired in-class scattering and it is necessary to remove them from the feature vectors, so that they are not used in the training of the decision tree. We call this process *feature preselection*. As a measure of scattering, we use standard deviation. A global threshold  $t$  is chosen to test it. Features with a standard deviation exceeding this threshold for at least one pattern type are removed from the feature set. In other words, in further processing, only features of index  $l$  that fulfill the below condition are kept:

$$\sqrt{\frac{1}{p_i} \sum_{j=1}^{p_i} \left( v_{i,j}^{(l)} - m_i^{(l)} \right)^2} < t, \quad \forall i = 1, \dots, k, \quad (5)$$

where  $v_{i,j}^{(l)}$  stands for  $l$ -th feature of the  $j$ -th example of  $i$ -th pattern class,  $m_i^{(l)}$  is the mean value of  $l$ -th feature in  $i$ -th pattern class. This guarantees that features used in the decision tree learning, chosen as split points for the decision tree present high in-class integrity and the classification result will not be influenced by common, slight distortions of the processed patterns. This is demonstrated in Fig. 6 showing two different shape classes, on features derived from the core boundary pixel class. Samples  $\mathcal{D}_{cb}^{(3)}$  and  $\mathcal{D}_{cb}^{(10)}$  were removed from the feature set due to excess scattering for shape “spider”, and sample  $\mathcal{D}_{cb}^{(7)}$  was removed due to excess scattering for shape “bird”. Therefore, the following subset of the initial samples of spatial class *core boundary* can be used in the decision tree growing:  $\{\mathcal{D}_{cb}^{(1)}, \mathcal{D}_{cb}^{(2)}, \mathcal{D}_{cb}^{(4)}, \mathcal{D}_{cb}^{(5)}, \mathcal{D}_{cb}^{(6)}, \mathcal{D}_{cb}^{(8)}, \mathcal{D}_{cb}^{(9)}, \mathcal{D}_{cb}^{(11)}\}$ . During the tree growing process, the split criteria are chosen for each node to maximize class integrity in child nodes after the division. There exist, however, some severe problems in decision

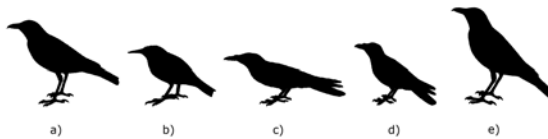


tree learning, such as *overfitting*, which can cause the tree to become overly complicated and inflexible. This phenomenon arises from the discrete and inherently greedy nature of the tree-growing algorithms, trying to properly include every data point into the tree structure, even if it is a statistically insignificant outlier. In our method, due to feature preselection and elimination of the most likely error-causing and outlier-influenced sections of the feature set, tree overfitting does not present itself as a problem, so it is possible to obtain full, unpruned trees with completely class-uniform leaves.

## 5 Results

Two rounds of testing were performed to evaluate the quality of classification. In the first round, a classification tree was constructed to perform recognition of binary shapes of 4 distinct pattern classes: “lizard”, “diamond”, “spider”, “bird” (shown in Fig. 3(a)). A training set of 160 binary patterns was used, where each pattern class was represented by 40 examples. The diversity of images within each pattern class was high, as shown in Fig. 7. It is worth pointing out, that the shapes differ from each other quite noticeably (compare the shape of the tail between (a) and (c)). Morphological classification, distribution function normalization and sampling were performed, with  $s_{CL} = 11$  samples for every spatial class  $CL$ . Since all patterns are represented by single connected components, class *isolated* was not taken into account – the feature vector was created based on the Eq. 4. The number of samples equal to 11 was enough to maintain the main characteristics of the class distribution function in the feature set at a satisfactory level. The preselection threshold for this round of testing was set to 0.035.

The tree-growing algorithm generated an output tree of size 4 and depth 3. This tree is shown in Fig. 9. The division points were chosen as samples  $v^{(6)}$ ,  $v^{(13)}$ , and  $v^{(38)}$ . The samples forming the feature vector  $\mathbf{v}$  are organized in the following manner (spatial classes and samples): *core* -  $v^{(1)}, \dots, v^{(11)}$ , *branches* -  $v^{(12)}, \dots, v^{(22)}$ , *core boundary* -  $v^{(23)}, \dots, v^{(33)}$ , *corridors* -  $v^{(34)}, \dots, v^{(44)}$ . Therefore, samples belonging to spatial classes: *core* ( $v^{(6)} \equiv \mathcal{D}_{cr}^{(6)}$ ), *branch* ( $v^{(13)} \equiv \mathcal{D}_{br}^{(2)}$ ) and *corridor* ( $v^{(38)} \equiv \mathcal{D}_{co}^{(5)}$ ) were chosen as decision tree splits. Tree testing was performed to verify the quality of the classifier. The test set consisted of 80 shapes, 20 for each of the shape types present in the training set



**Fig. 7.** Sample shapes for feature set generation: “bird1” (a), “bird2” (b), distorted (c) and undistorted (d) shape “bird3”, distorted “bird1” (e)

(“lizard”, “diamond”, “spider”, “bird”). The images were chosen basing on the same criteria as with the training set: there were different natural shapes and shapes derived from these natural shapes by means of scaling and rotation. This set was tested against the obtained classification tree, and a 100% classification accuracy was achieved. In this test scenario, the best class predictors turned out to be the *core*, *corridors* and *branch* spatial classes. The distribution functions for these pixel classes show a high diversity between the shapes, and their characteristics are highly distinctive. It is therefore feasible to perform a pre-analysis on smaller sets and construct a feature set consisting only of a subset of pixel class samples to achieve good classification results for a specific pattern composition.

A second test was performed on a larger set of shapes. This time patterns was belonging to 8 classes: “bird”, “spider”, “diamond”, “lizard”, “plane”, “octopus”, “hand” and “whale” (shown in Fig. 8).



Fig. 8. Sample shapes used for the second round of testing

The training set consisted of 320 images, where each class was represented by 40 images derived from the basic shape by means of scaling and rotation to achieve high in-class diversity. The test set consisted of 160 images, with 20 images representing each of the 8 shape classes. Distribution function calculation, feature extraction, preselection and tree growing were performed as in the first round of testing, with  $s_{CL} = 11$  samples for every spatial class  $CL$ . The preselection threshold was set to 0.135. Again, high accuracy of classification was achieved, with 98.75% of shapes being properly attributed to their shape class (two objects were improperly classified). The decision tree generated in the second round of testing is presented in Fig. 10. This time, samples belonging to the following spatial classes were used as decision tree splits: *core* ( $v^{(1)} \equiv \mathcal{D}_{cr}^{(1)}$ ,

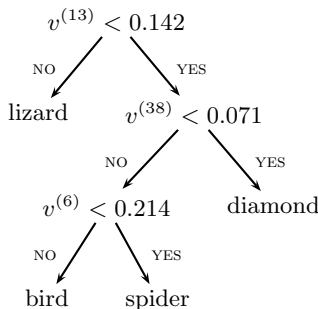
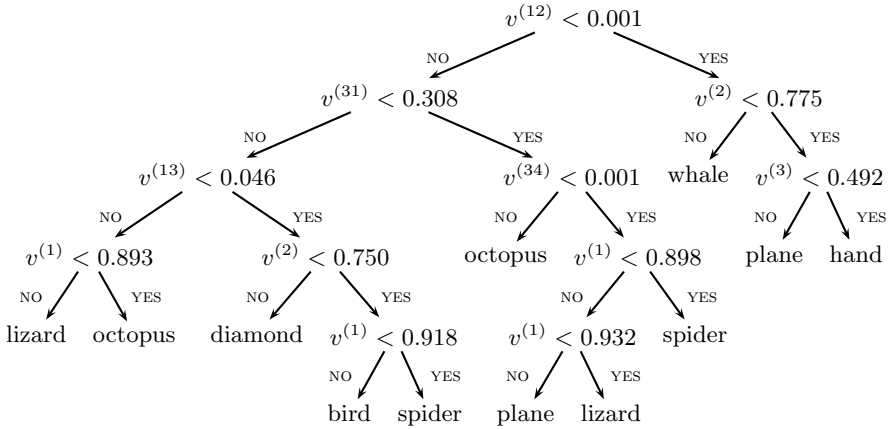


Fig. 9. Decision tree structure with marked split points



**Fig. 10.** Decision tree structure generated in the second round of testing

$v^{(2)} \equiv \mathcal{D}_{cr}^{(2)}$ ,  $v^{(3)} \equiv \mathcal{D}_{cr}^{(3)}$ , *branch* ( $v^{(12)} \equiv \mathcal{D}_{br}^{(1)}$ ,  $v^{(13)} \equiv \mathcal{D}_{br}^{(2)}$ ), *core boundary* ( $v^{(31)} \equiv \mathcal{D}_{cb}^{(9)}$ ) and *corridor* ( $v^{(34)} \equiv \mathcal{D}_{co}^{(1)}$ ). Similarly to the first test, in the second test scenario the shape classification was determined most heavily by the *core* and *branch* spatial class distributions.

## 6 Conclusions

In the paper, the method for classifying binary patterns was proposed. It consists of several steps and starts with classification of pixels belonging to binary patterns into several spatial classes, which is performed using morphological image processing. By performing this classification with structuring elements of increasing sizes, the spatial class distribution functions are produced. These functions are normalized and sampled in order to obtain feature vectors of constant length that are invariant to translation, rotation and scaling of binary pattern. Such feature vectors are next used to perform decision-tree classification. Prior to proper decision-tree classification an additional step of feature preselection is performed based on the training set. It allows removing from this feature set features with high intra-class scattering, and in effect, makes the remaining features more suitable for pattern class separation.

The tests confirm that the proposed method is a robust and effective tool for binary pattern recognition. Furthermore, the method shows a level of flexibility and, if required, can be optimized for a specific set of shapes to achieve better performance and match accuracy.

As pixel class extraction from pictures containing the binary shapes is the most time-consuming step of the classification task, in most cases it is possible to narrow the analysis to a subset of pixel classes (for example, in the test scenario presented in this paper, only three pixel classes proved relevant). Moreover,

since pixel class extraction is a task that is limited in dependencies to the individual picture processed, it can be performed in parallel by multiple instances of the extracting program, running on a multi-core CPU or multi-processor machine. Morphological operations performed during feature extraction can also be parallelized to a large extent, allowing for further reduction of processing time on modern computer hardware.

## References

1. Soille, P., Vogt, P.: Morphological segmentation of binary patterns. *Pattern Recognition Letters* 30 (2009)
2. Iwanowski, M.: Morphological Classification of Binary Image Pixels. *Machine Graphics and Vision* 18 (2009)
3. Riitters, K., Vogt, P., Soille, P., Kozak, J., Estreguil, C.: Neutral model analysis of landscape patterns from mathematical morphology. *Landscape Ecology* 22, 1033–1043 (2007)
4. Iwanowski, M.: Binary Shape Characterization using Morphological Boundary Class Distribution Functions. In: *Advances in Intelligent and Soft Computing - Computer Recognition Systems 2*. Springer, Heidelberg (2007)
5. Vogt, P., Riitters, K., Iwanowski, M., Estreguil, C., Kozak, J., Soille, P.: Mapping Landscape Corridors. *Ecological Indicators* 7(2) (2007)
6. Soille, P.: *Morphological image analysis*. Springer, Heidelberg (1999) (2004)
7. Serra, J., Vincent, L.: An overview of morphological filtering. *Circuit Systems Signal Processing* 11(1) (1992)
8. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and Regression Trees*. CRC Press, Boca Raton (1984)
9. Serra, J.: *Image analysis and mathematical morphology*, vol. 1. Academic Press, London (1983)