Ontology Based Complex Object Recognition

Nicolas Maillot and Monique Thonnat

INRIA Sophia Antipolis - Orion Team
2004 Route des lucioles - B.P. 93
06902 Sophia Antipolis, France
{Nicolas.Maillot, Monique.Thonnat}@sophia.inria.fr

Abstract

This paper presents an object categorization method. Our approach involves the following aspects of cognitive vision : machine learning and knowledge representation. A major element of our approach is a visual concept ontology composed of several types of concepts (spatial concepts and relations, color concepts and texture concepts). Visual concepts contained in this ontology can be seen as an intermediate layer between domain knowledge and image processing procedures.

This paper details this approach which is composed of three phases: a knowledge acquisition phase, a learning phase and a categorization phase. A major issue is the symbol grounding problem (symbol grounding consists in linking meaningfully symbols to sensory information). We propose a solution to this difficult issue by showing how learning techniques can map numerical features to visual concepts.

Key words: Ontology, Machine Learning, Categorization, Cognitive Vision

1 Introduction

This paper presents a cognitive vision approach designed for complex object categorization. Both knowledge representation and machine learning techniques are involved in the categorization process. The proposed approach is designed for semantic interpretation of isolated objects of interest. Related work on scene analysis issues (i.e. involving non isolated objects) can be found in (1).

A long experience in complex object categorization (2) (3) has shown that experts often use a well defined vocabulary for describing the object of their domain. Based on that statement, results coming from the knowledge engineering community can be applied to acquire expert knowledge. This expert knowledge is used to guide object recognition.

This approach raises the question of relations between language and perception. In (4), Xu explains that language is a way to develop object discrimination capabilities. Our work is inspired by this claim : we aim at using a conceptualization of the visual perception domain (a visual concept ontology) to define object recognition strategies. Moreover, our approach can be brought close to the procedural semantics (5) theory where visual concepts and their labels are associated in the sense that they are alternative ways of gaining access to same underlying procedures (segmentation, feature extraction and learning modules).

Section 2 gives an overview of key issues and existing approaches in high-level image interpretation. Section 3 gives a global point of view on the proposed approach. Section 4 details the structure of a priori knowledge (i.e. domain knowledge and visual concept ontology) involved the recognition process. Section 5 is dedicated to image processing techniques used for achieving categorization. Section 6 explains how visual concepts are learnt by learning techniques. Section 7 presents an object learning algorithm and an object categorization algorithm. A discussion on our approach is given in section 8. We finally conclude in section 9.

2 Related Work

In (6), an introduction to high-level bayesian image interpretation techniques can be found. The author explains that bayesian analysis techniques are more widely applicable and reliable that ad hoc algorithms. Such statistical models are explicit and allow to evaluate confidence about conclusions. The difficult task in the construction of bayesian models is to define prior distribution. In particular, context (e.g. point of view, scale, acquisition conditions) is hard to manage. That is why care is required in using statistical knowledge effectively for a given specific problem.

A solution to invariance problems are local invariant image feature detectors (7) (8). The main drawback of these techniques is that they are well adapted to object (i.e. an instance of a given class) recognition and not to class recognition. Preliminary work on the generalization of these techniques to class recognition is starting (9).

Knowledge based vision systems have proven to be effective for complex object recognition (2) and for scene understanding (10). They offer a great capacity of reusability and extendability. Moreover, in knowledge based systems, domain knowledge is clearly separated from image processing knowledge. This implies a better tractability of the different sub-problems (i.e. image processing and interpretation) encountered in image understanding. The major negative point of these systems is that they rely on knowledge bases which are difficult to produce and manage.

To achieve complex object recognition, we propose an intermediate approach: to use a priori knowledge to structure prior distributions of relevant visual features (i.e. texture, color, shape). A priori knowledge is used to perform a focused learning of distinctive visual characteristics.

3 Proposed Approach

Our approach is composed of three main phases : (1) a knowledge acquisition phase, (2) a learning phase, (3) a categorization phase. This section gives an overview on the whole approach.

3.1 Knowledge Acquisition Phase

First comes knowledge acquisition issues which have been discussed in (11). A dedicated knowledge acquisition tool has been implemented and allows to perform the following tasks:

- domain taxonomy acquisition (i.e. hierarchy of domain classes)
- ontology driven visual description of domain object classes which leads to a more complete domain knowledge base
- image sample management (i.e. annotation and manual segmentation of samples of object classes of interest)

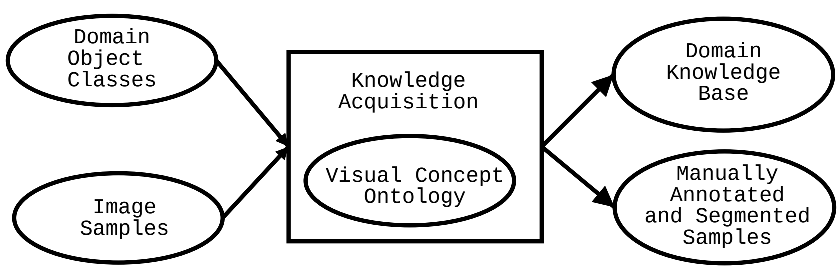

Fig. 1. Knowledge Acquisition Phase Overview

As seen in fig. 1, the knowledge acquisition process leads to a knowledge base in which a set of domain classes are described by visual concepts. Manually segmented and annotated image samples of domain objects are also obtained. Sample annotation consists in labeling a set of sample images by a domain object class name.

Expert knowledge is acquired during the knowledge acquisition process. A remaining important issue is the symbol grounding problem (12). The learning phase fills the gap between symbols used during knowledge acquisition and manually segmented and annotated sample images. As seen in fig. 2, three modules are involved in the object learning process.

(1) The main module is the object learning module which controls the other modules. The learning process is initiated by a learning request which contains a list of classes of the domain taxonomy. For specific applications, some classes are not relevent. Therefore, it may be needed to restrict learning process to a subpart of the whole domain knowledge. This module first retrieves domain class samples (i.e. a set of regions of interest annotated by a domain class instance).
(2) The feature extraction module accepts feature extraction requests sent by the object learning module and computes features and segmented samples.
(3) The visual concept learning module trains a set of classifiers by using features extracted by the feature extraction module. These classifiers are trained to the recognition of visual concepts used for the description of domain classes. The output of the object learning module is a knowledge base augmented with the trained classifiers.

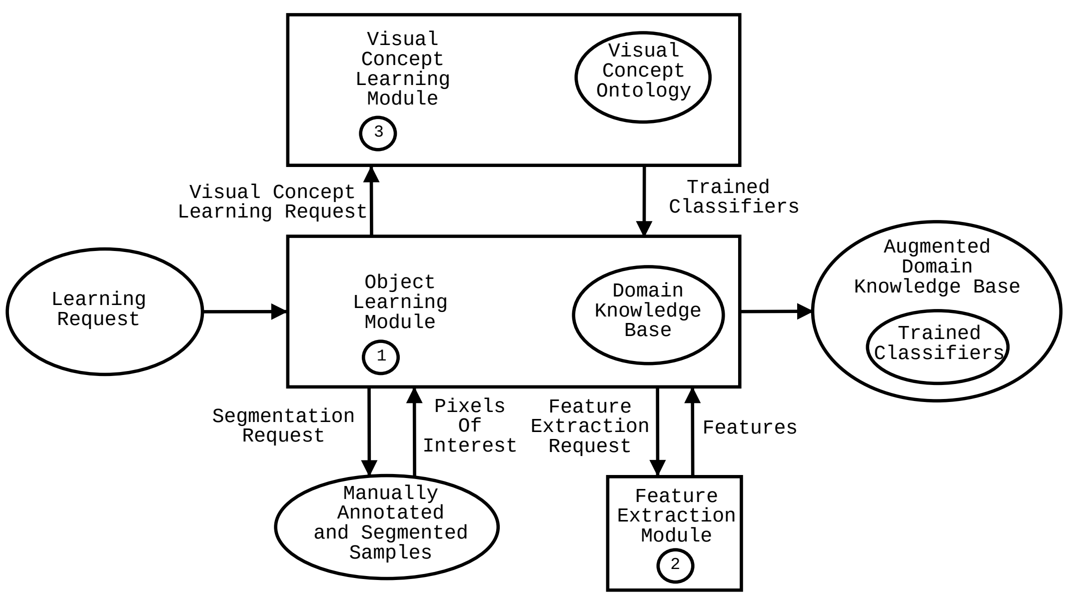

Fig. 2. Object learning phase overview

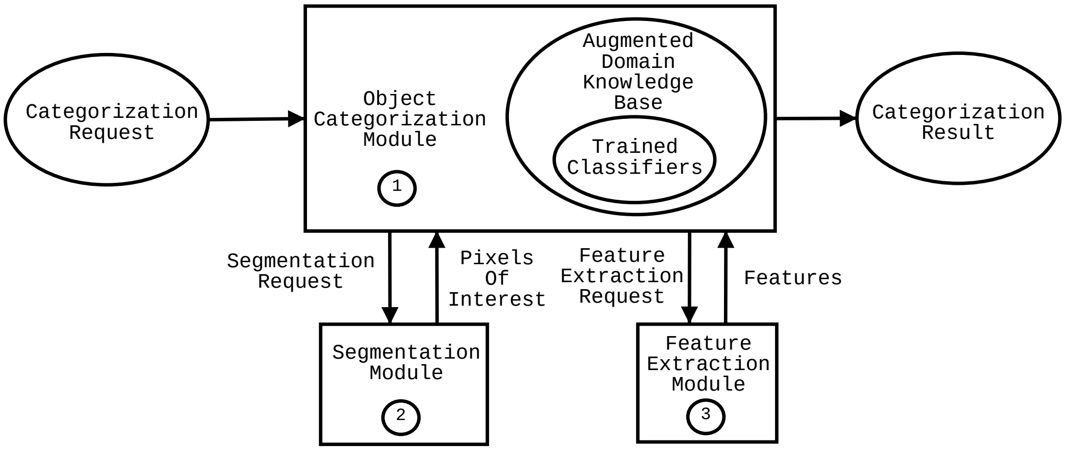

Fig. 3. Object categorization phase overview

Fig. 4. Image of an isolated pollen grain

3.3 Categorization Phase

Fig. 3 gives an overview of the proposed object categorization phase. This phase is based on three modules:

(1) The object categorization module is the central module of the categorization architecture. It processes categorization requests so as to produce categorization results. A categorization request is composed of an image containing an isolated object of interest (see fig. 4). Information concerning context (e.g. acquisition device, date) is also integrated in the categorization request. To perform object categorization, the object categorization module sends sequentially two different types of requests to the segmentation module and to the feature extraction module. Recognition of visual concepts is done by using classifiers trained during the learning phase and contained in the augmented knowledge base. Categorization result contains one or several domain object classes which match recognized visual concepts associated with the current object to be recognized.

(2) The segmentation module receives segmentation requests. The answer to a segmentation request is a set of pixels of interest inside a region of interest. A segmentation request is composed of the region of interest where pixels of interest have to be extracted. A symbolic description (in terms of visual concepts) of the expected segmentation result is also integrated in the request.

(3) The feature extraction module accepts feature extration requests sent by the object categorization module. The feature extraction module transforms segmented pixels of interest into numerical features (e.g. Gabor features for texture analysis).

4 A Priori Knowledge

4.1 A Visual Concept Ontology

As defined in (13), an ontology is a formalization of a conceptualization. An ontology defines the semantics of non-logical primitives used in a knowledge representation language. An ontology is composed of :

- a set of concepts (C) (e.g. geometric concepts)
- a set of relations(R) (e.g. spatial relations)
- a set of axioms (e.g. transitivity, reflexivity, symmetry of relations)

Two partial orders $\preceq_C$ and $\preceq_R$ define the concept hierarchy and the relation hierarchy, respectively. An ontology is supposed to be the support of reasoning mechanisms.

In this section, we propose a visual concept ontology. This ontology can be considered as a guide which provides a vocabulary for the visual description of domain classes. It is important to note that the proposed ontology is not application-dependent and should be considered as an extendable basis. We have structured this ontology in three main parts. The first one contains texture concepts, the second one contains color concepts and the last one is made of spatial concepts. Each part of this ontology is detailed in the next subsections.

4.1.1 Texture Concepts

This part of the ontology has been inspired by results from the cognitive science community.

The experiment conducted in (14) identifies three main dimensions in the texture perception cognitive process. Each perceptual dimension constitutes an important element in texture perception. Each perceptual dimension is seen as an abstraction of a set of visual concepts (Fig. 5). From this study, we have built an ontology of texture concepts. Note that quantifiers (e.g. *Non*, *Average*, *Strong*) are also integrated in this ontology and can be used to give a finer visual description.

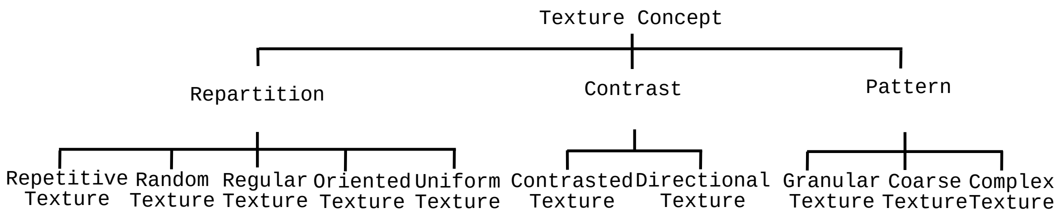

Fig. 5. Texture concept hierarchy

4.1.2 Color Concepts

This part of the ontology is derived from the ISCC-NBS (Inter-Society Color Council-National Bureau of Standards) color dictionnary. An interesting reflexion on the validity of this dictionnary is given in (5). Three kinds of notions are included: hue, brightness and saturation concepts. There are 28 hue concepts (Table 1) which can be combined with five brightness concepts (*Very Dark, Dark, Medium, Light, Very Light*) and four saturation concepts (*Grayish, Moderate, Strong, Vivid*). Certain combinations of brightness and saturation concepts have a perceptual meaning. For instance, the concept *Brillant* is an association of the *Light* and *Strong* concepts. Axioms are contained in the ontology so as to express those kinds of associations.

4.1.3 Spatial Concepts

This part of the ontology is used for describing domain objects from a spatial point of view. A part of the hierarchy is composed of geometric concepts that can be used to describe the shape of domain objects (fig. 6). Three other parts of the ontology contains position,orientation and size concepts. A formalization of a similar approach based on a combination of geometric shapes can be found in (15). The size of an object can also be described and quantified with a set of quantifiers. Note that quantification can be done in an absolute way or relatively to another concept. This means that the size of object A can be described as being important relatively to object B. The notion of elongation is also present and can be quantified. We have also added a set of spatial relations based on the RCC-8 model (16) that can be used to define relations between objects and their subparts.

These relations are enumerated in Table 2 and graphically represented in fig.

Red	Purple
Reddish Orange	Reddish Purple
Orange	Purplish Red
Orange Yellow	Purplish Pink
Yellow	Pink
Greenish Yellow	Yellowish Pink
Yellow Green	Brownish Pink
Yellowish Green	Brownish Orange
Green	Reddish Brown
Bluish Green	Brown
Greenish Blue	Yellowish Brown
Blue	Olive Brown
Purplish Blue	Olive
Violet	Olive Green

Table 1
Set of hue concepts

7.

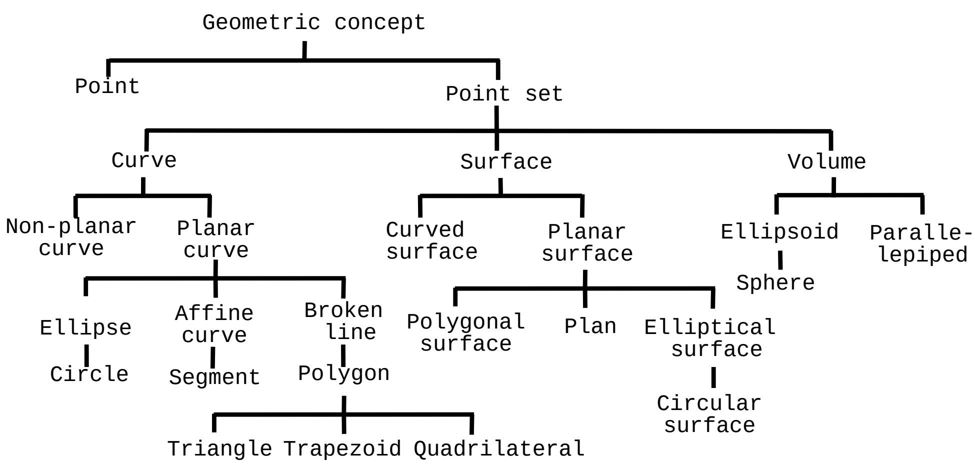

Fig. 6. Geometric concept hierarchy

4.1.4 Context Description

Experts often observe the objects of their domain in precise observation conditions. For example, when using a microscope, magnification or lighting conditions are controled. Providing contextual information is absolutely necessary. Context information is the link between domain knowledge and image

RCC-8 relation	Meaning
DC(X,Y)	X disconnected from Y
EC(X,Y)	X externally connected to Y
EQ(X,Y)	X equals Y
PO(X,Y)	X partially overlapping Y
TPP(X,Y)	X tangential proper part of Y
TPP-1(X,Y)	X has tangential proper part Y
NTPP(X,Y)	X nontangential proper part of Y
NTPP-1(X,Y)	X has nontangential proper part Y

Table 2
RCC-8 relations and their meaning

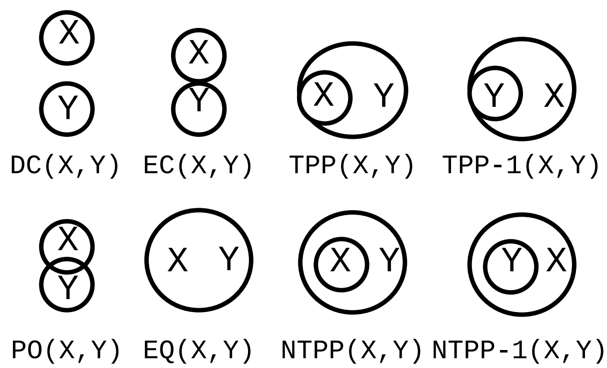

Fig. 7. RCC-8 graphical representation

samples resulting from the acquisition process. Context conditions the resulting acquired images. This implies a relation between the visual description of image samples and the context of acquisition. Context knowledge avoids building incoherent sets of image samples. For instance, it would not make sense to gather images of a similar object acquired with different types of sensors. Context depends on the application domain. That is why the context hierarchy given in fig. 8 can be extended and adapted for a particular domain.

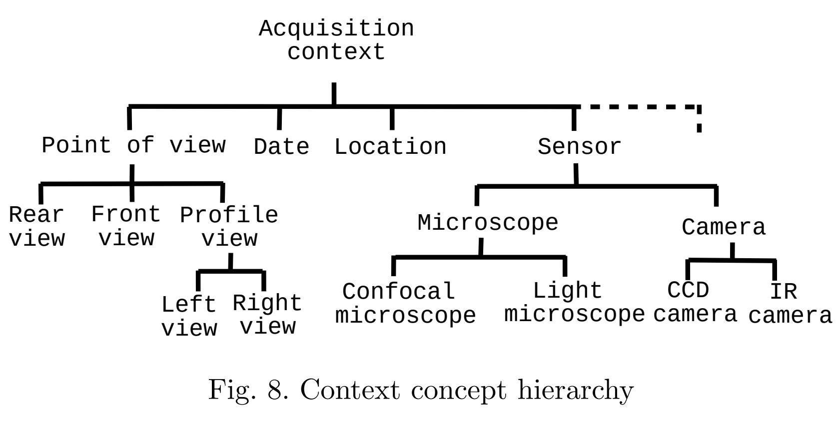

Fig. 8. Context concept hierarchy

This knowledge belongs to the domain of interest and is shared by the specialists of the domain (e.g. biologists, astronomers). Domain knowledge is independant of any vision layer and can be reused for other purposes. Domain knowledge is structured as a hierarchy of classes (i.e. a taxonomy). As seen in fig. 9, composition links allow to define part-whole relations between domain classes.

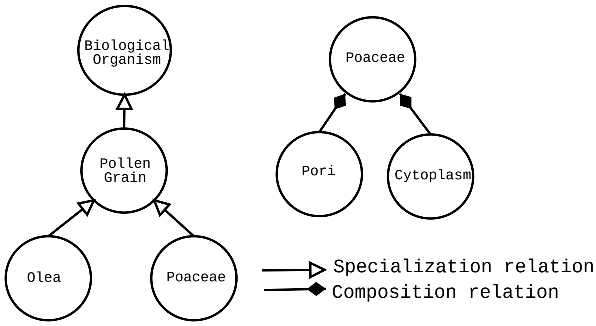

Fig. 9. Domain knowledge structure: on the left, a taxomony, on the right a partonomy

An example of a domain class formalized with the frame formalism can be found in table 3. The **SuperClass** attribute allows to define specialization relations. The **SubParts** attribute defines composition relations. There are three categories of attributes related to visual description:

(1) spatial attributes : geometry; size; orientation; position
(2) color attributes : hue; brightness; saturation
(3) texture attributes : repartition; contrast; pattern

Each attribute has a defined type and a set of possible values. For instance, attribute **hue** is of type *HueConcept* and has for value *Pink*. The relation between possible values and attribute types is a specialization relation.

5 Image Processing

5.1 Segmentation

Segmentation needs have been introduced in section 3. The goal of this paper is not to give much details on this specific problem. Let us consider that the segmentation module is able two perform different types of segmentation tasks. The first segmentation task is isolation of the object of interest from

Class	POACEAE
{	
SuperClass:	POLLENWITHPORI
SubParts:	
PORI pori1	[PORIWITHANULUS]
SpatialAttributes :	
GeometricConcept **geometry :**	[*CircularSurface EllipticalSurface*]
SizeConcept **size** :	[*ImportantSize*]
ColorAttributes :	
HueConcept **hue**:	[*Pink*]
BrightnessConcept **brightness**:	[*Dark*]
TextureAttributes :	
TexturePatternConcept **pattern**:	[*GranulatedTexture*]
TextureContrastConcept **contrast**:	[*Slight*]
SpatialRelations :	
SpatialRelation **r1**:	[*NTTP*(POACEAE,pori1) *TTP*(POACEAE,pori1)]
}	

Table 3
High level description of domain class poaceae. Visual concepts provided by the ontology are in *italic*. Attribute names are in **bold face**. Knowledge provided by the expert is in SMALL CAPS.

the background. The second segmentation task is the extraction of object subparts.

A segmentation request is composed of two elements: a list of visual concepts and the region of interest where segmentation is performed. Segmentation is guided by visual concepts included in the segmentation request. The reader can refer to (2) and (17) to understand how program supervision techniques are used to control and reuse image processing algorithms.

5.2 Feature Management and Extraction

Section 4 has introduced the structure of the proposed visual concept ontology. During the classification of a given object, numerical features are computed. To be interpreted as visual concepts, a link must be established between computed numerical features and symbolic visual concepts. As explained in sub-

Class	*GeometricConcept*
{	
Superclass:	*SpatialConcept*
GeometricFeaturesAttributes:	
Float length	$[0, +\infty]$
Float width	$[0, +\infty]$
Float lengthWidthRatio	$[0, +\infty]$
Float area	$[0, +\infty]$
Float formFactor	$[0, +\infty]$
Float perimeter	$[0, +\infty]$
Float roundness	$[0, +1]$
Float compactness	$[0, +1]$
...	
}	

Table 4
An Example of the Visual Concept *GeometricConcept*. Some geometric features are given. These features are used during the visual concept learning process. Restrictions on the domain of the features are also defined.

section 4.2, several categories of visual concepts are managed: geometry; size; orientation; position; hue; brightness; saturation; repartition; contrast; pattern. We propose to create a sets of numerical features associated with these visual concepts.

Numerical features depend on available feature extraction algorithms. It is up to the image processing expert to make relevent association between available feature extraction algorithms and visual concepts. As seen in table 4, the concept *GeometricConcept* can be represented in a frame formalism. Attributes of a given visual concept are inherited by its subconcepts (e.g *CircularSurface*).

Visual concepts feature attributes are computed by the feature extraction module. For instance, numerical features associated with the concept *TextureConcept* are statistical moments and Gabor Features. Color characterization is done with histograms and color coherence vectors (18). These features are used to learn and recognized visual concepts, this learning process is described in the next section.

6 Visual Concept Learning

6.1 Overview

Although visual concept semantics is provided by the visual concept ontology, symbol grounding remains an issue. The visual concept learning module is used to fill the gap between ontological concepts and image level. As explained in fig. 10, the visual concept learning module learns a set of classifier to the recognition of each visual concept. This learning is done thanks to a set of training vectors computed by the feature extraction module computed on manually segmented and annotated regions of interest. These features vectors are included in the visual concept learning request. The visual concept ontology is used in this module because the learning process is done in a hierarchical way by using the ontological tree structure. Learning has to be focused on specific problems. Therefore, a visual concept is also included in the request so as to define the subpart of the ontology which has to be learnt. For instance, if the visual concept *GeometricConcept* is inside the request, only this concept and its children are involved in the learning process.

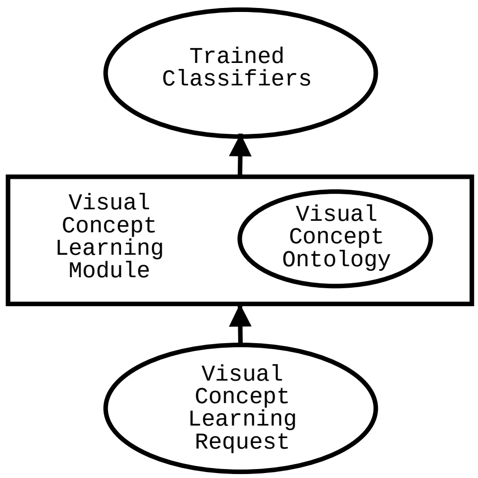

Fig. 10. From a set of labeled regions to a set of trained classifiers

6.2 Problem Statement

The proposed architecture is designed to learn a set of visual concepts used during knowledge acquisition. A training set S_i is associated with each visual concept $C_i \in C$. A training set is a set of N labeled vectors $\mathbf{x_i} \in \mathbf{R}^n$ computed by the feature extraction module described in section 3. Vectors are labeled by $y_i \in \{-1, 1\}$. $y_i = 1$ means that $\mathbf{x_i}$ is a representative sample of C_i. $y_i = -1$ means that $\mathbf{x_i}$ is a negative sample of C_i. More precisely, negative samples of C_i

are positive samples of brothers of C_i. This means that we use the hierarchical structure of the ontology to obtain simpler and focused classification problems.

A classifier d_i is associated to each concept C_i (see table 5). $P(C_i|\mathbf{x})$ is the posterior probability of C_i. $P(\neg C_i|\mathbf{x})$ is the posterior probability of $\neg C_i$. We also introduce a reject distance class R_i so that $P(C_i) + P(\neg C_i) + P(R_i) = 1$. This distance reject class allows to take into account vectors observed in unexpected regions of $\mathbf{R}^n$. The probability law of $\mathbf{x}$ is defined as $p(\mathbf{x}) = p(\mathbf{x}|C_i)P(C_i) + p(\mathbf{x}|\neg C_i)P(\neg C_i) + p(\mathbf{x}|R_i)P(R_i)$.

We define two thresholds $\alpha_{amb} \in]0.5, 1[$ and $\alpha_{dist} \in]0, 1[$. α_{amb} is the ambiguity reject threshold and defines the degree of confidence needed to take the decision of recognizing a concept. α_{dist} is the distance reject threshold. Distance reject is inferred from $p(\mathbf{x})$ and α_{dist} and classifies $\mathbf{x}$ into R_i. Distance reject means that $\mathbf{x}$ is unlikely to belong to both C_k and $\neg C_k$ and might belong to a concept that has not been learnt yet. For more details about the distance reject notion, see (19).

$d_i(\mathbf{x})$	Definition
C_i recognized	$P(C_i|\mathbf{x}) \geq \alpha_{amb}$
C_i not recognized	$P(\neg C_i|\mathbf{x}) \geq \alpha_{amb}$
Ambiguity reject	$max\{P(C_i|\mathbf{x}), P(\neg C_i|\mathbf{x})\} < \alpha_{amb}$
Distance reject	$p(\mathbf{x}) < \alpha_{dist}$

Table 5
Decision types

The following subsections aim at presenting a methodology designed to build each classifier d_i. Each classifier is trained to classify the training vectors $\mathbf{x_i}$ labeled by y_i such that $y_i = 1$. This training is done against training vectors x_j labeled by $y_j = -1$. This approach is a one-versus-rest classification scheme. This means that each C_i is learnt and classified against all its brothers.

The learning module is made of three sub-modules : a training set building module, a feature selection module and a training module (Fig. 11). Each of these modules is detailed in the next subsections. The main algorithm is given in algorithm 1. This algorithm is executed when a visual concept learning request is sent to visual concept learning module. It sequentially calls the training set building module, the feature selection module and the training module. This algorithm takes two parameters. The first one is a set of feature vectors labeled by visual concepts. The second one is a visual concept C_i (e.g. *GeometricConcept, HueConcept*) which defines a subpart of the ontology. The result of algorithm 1 is a set of classifiers trained to the recognition of C_i and its children.

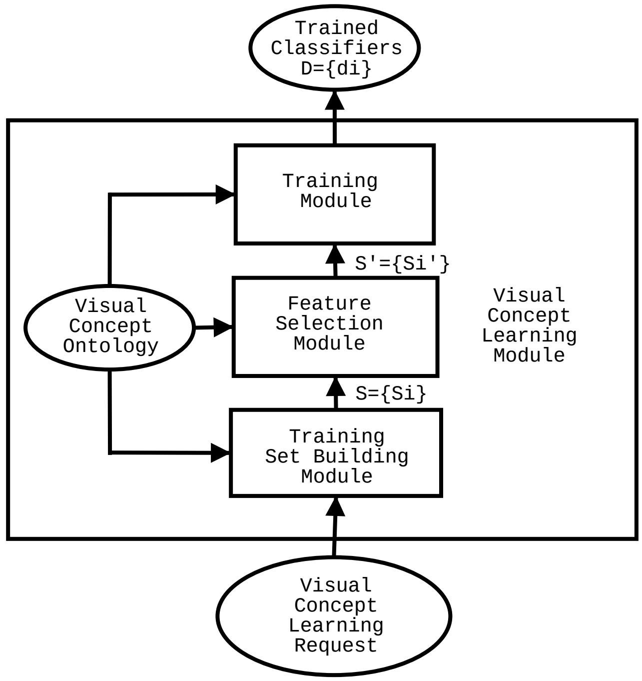

Fig. 11. Learning module

Algorithm 1 $VisualConceptLearning(X, C_i)$

$TrainingSetBuilding(X, C_i, S)$
$FeatureSelection(C_i, S, S')$
$Training(C_i, S', D)$
$return\ D$

6.3 Training Set Building Module

As explained in previous subsections, each visual concept used during knowledge acquisition has an attached classifier. The first step is to build training sets to train classifiers associated with visual concepts. For each visual concept used during visual description, image processing algorithms compute a set of feature vectors. Regions of interest are used as an input of the feature extraction module.

As shown in Fig. 11, the visual concept learning module is designed to process a visual concept learning request. The set of training vectors contained in a visual concept learning request ($X = \{\mathbf{x_i}, \mathbf{C_i}\}$) is computed by the feature extraction module (see Fig. 2). Vector labels are provided by the ontology driven description of domain classes. Note that two similar vectors can be labeled by several visual concepts: we are in a multi-label case. The labeled vectors are then processed by the training set building module. P_i is the set of representative training vectors of a visual concept C_i. N_i is the set of training vectors computed on negative samples of a visual concept C_i. The training set associated with C_i is noted S_i. The training set building module aims at computing S_i (feature vectors labeled by $+1$ or -1) for each C_i. This implies that the hierarchical structure of the ontology has to be used to compute each S_i.

$$\begin{cases} P_i = \bigcup_j \{(\mathbf{x_j}, +1) \mid C_j \preceq_C C_i\} \\ N_i = \bigcup_j \{(\mathbf{x_j}, -1) \mid C_j \preceq_C (C_k \in brothers(C_i)) \wedge (\mathbf{x_j}, +1) \notin P_i\} \\ S_i = P_i \cup N_i \end{cases}$$

6.4 Feature Selection Module

Feature selection is hierarchically performed by Algorithm 2. *FeatureSelection-Algorithm* function applies a feature selection algorithm to each S_i to obtain each Si'. We currently use a Sequential Forward Floating Selection (SFFS) Algorithm (20). This method iteratively adds or removes features until some termination criterion is met. Bhattacharyya distance (21) between classes is used as a separability criterion. This implies that features used to recognize a visual concept may be different from features used for the recognition of another concept.

Algorithm 2 $FeatureSelection(C_i, S, S')$

```
children ← getChildren(C_i)
for all C_j ∈ children do
  S'_j ← FeatureSelectionAlgorithm(S_j)
  S' ← S' ∪ S'_j
  if hasChildren(C_j) = true then
    FeatureSelection(C_j, S, S')
  end if
end for
```

6.5 Training Module

As detailed in Algorithm 3, the learning process is guided by the hierarchical structure of the ontology. Algorithm 3 is initially called with a visual concept C_i such as *GeometricConcept* and $S' = \{S'_i\}$. The set of classifiers $D = \{d_i\}$ is built recursively. Learning of each descendant of C_i is performed hierarchically. The *trainClassifier* function first loads S_i, then creates and trains a binary classifier d_i to the recognition of C_i. We currently use multi layer perceptrons and k nearest neighbors as binary classifiers. Next section shows how $\{di\}$ is used to learn and recognize domain classes.

Algorithm 3 $Training(C_i, S', D)$

$children \leftarrow getChildren(C_i)$
for all $C_j \in children$ **do**
  $d_j \leftarrow TrainClassifier(S'_j)$
  $D \leftarrow D \cup d_j$
  if $hasChildren(C_j) = true$ **then**
    $Training(C_j, S', D)$
  end if
end for

6.6 Results

This subsection illustrates an application of the classifiers trained by the visual concept learning module. A cognitive experiment has been performed in (14) : a subset of 56 Brodatz texture images has been given to 20 persons who were asked to classify them in different clusters. A few samples of the Brodatz texture set are given in fig. 12. The clusters were formed by evaluating the following symbols (between 1 and 9): contrast, repetitveness, granularity, randomness, roughness, density, directionality, complexity, coarseness, regularity, orientation.

To perform the learning process described in this section, we have used texture related visual concepts associated with each Brodatz image. Each image is splitted in 16 pieces so as to obtain 128x128 images. Available texture image analysis algorithms (i.e. Gabor filters, auto-correlation matrices and cooccurence matrices) have been applied to obtain a training set. This training set has been processed by the training set building module. The feature selection module has reduced the number of features from 127 to 20. The total number of training vectors is 896 (56x16). Classification results presented in table 6 have been obtained by *N-fold cross-validation* (N=56). This evaluation approach consists in dividing the training set in N subsets. Then, feature selection, training, and classification are repeated N times. At each step, a subset is selected and used for obtaining classification results. The remaining N-1 subsets are used by the learning module. Results are the average of the N classification results obtained by using multi layer perceptrons classifiers with ambiguity reject activated. Classification results at the intermediate level of the texture ontology (i.e. *Repartition, Contrast, Pattern*) allow to see how main distinctive properties of image samples are recognized.

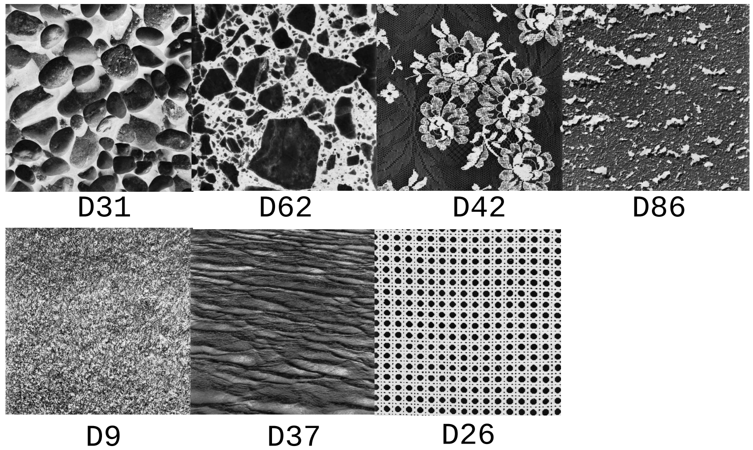

Fig. 12. Brodatz texture samples

Concept	False Positive	False Negative	True Positive
Repartition	2.81%	0.2%	99.8%
Contrast	19.9%	27.3%	72.7%
Pattern	21.9%	24.2%	75.8%
Repetitive	25.3%	6%	94%
Random	17.8%	24.1%	75.9%
Regular	23.9%	13.1%	86.9%
Oriented	7.8%	2.2%	97.8%
Uniform	23.3%	1.2%	98.8%
Directional	19.9%	27.3%	72.7%
Granular	8.8%	0%	100%
Non Granular	22.1%	29.1%	70.9%
Non Repetitive	21.2%	36.9%	63.1%
Non Random	12.7%	26.5%	73.5%

Table 6
Mutli Layer Perceptron based classification results

7 Object Learning and Categorization Algorithms

7.1 Object Learning Algorithm

As seen in fig. 13, learning process of domain classes is composed of three main steps. This algorithm is located in the object learning module (section

3) and learns the visual concepts used as values of domain classes attributes (e.g. size, hue, pattern).

(1) The first step consists in getting positive and negative samples of the domain class which has to be learnt. To perform this task, manually segmented and annotated samples are used.
(2) The second step consists in extracting features associated with attribute values (e.g. C_i =*CircularSurface*). Feature extraction is performed by the feature extraction module. For each attribute value, a set of positive feature vectors is obtained. The set of negative training vectors is labeled by $not(C_i)$. A recursive call is needed in order to learn the description of subparts of the current class.
(3) Third comes visual concept learning which has been described in the previous section. Algorithm 1 is called for each category of visual concept (i.e. spatial attributes : geometry, size, orientation, position; color attributes: hue, brightness, saturation; texture attributes: repartition, contrast, pattern).

7.2 Object Categorization Algorithm

Object categorization algorithm structure is given in fig. 14. This algorithm is divided in five steps. It tries to match an unknown object to be categorized with one or several classes of the domain. Matching is first performed at a local level. This local matching consists in comparing expected visual concepts and visual concepts computed on the unknown object to be classified. The global matching consists in combining the local matchings performed at the local level.

(1) The categorization process is initiated by a categorization request which contains an image of the object to be classified.
(2) Object of interest has to be segmented from background. If the algorithm tries to classify a subpart, the segmentation task consists in extracting the subpart from the main object. In both cases, a segmentation request has to be sent to the segmentation module.
(3) Then comes local matching between current class attribute values (e.g. *CircularSurface* for attribute **geometry**) and visual concepts recognized by the classifiers trained during the learning process. Features used for visual concept recognition are provided by the feature extraction module. The result of local matching is a set of probabilities associated with each attribute value. If the attribute is a subpart, a recursive call has to be made so as to categorize it.
(4) This step consists in evaluating if current class matches the object to be recognized. This matching is done by combining probabilities computed

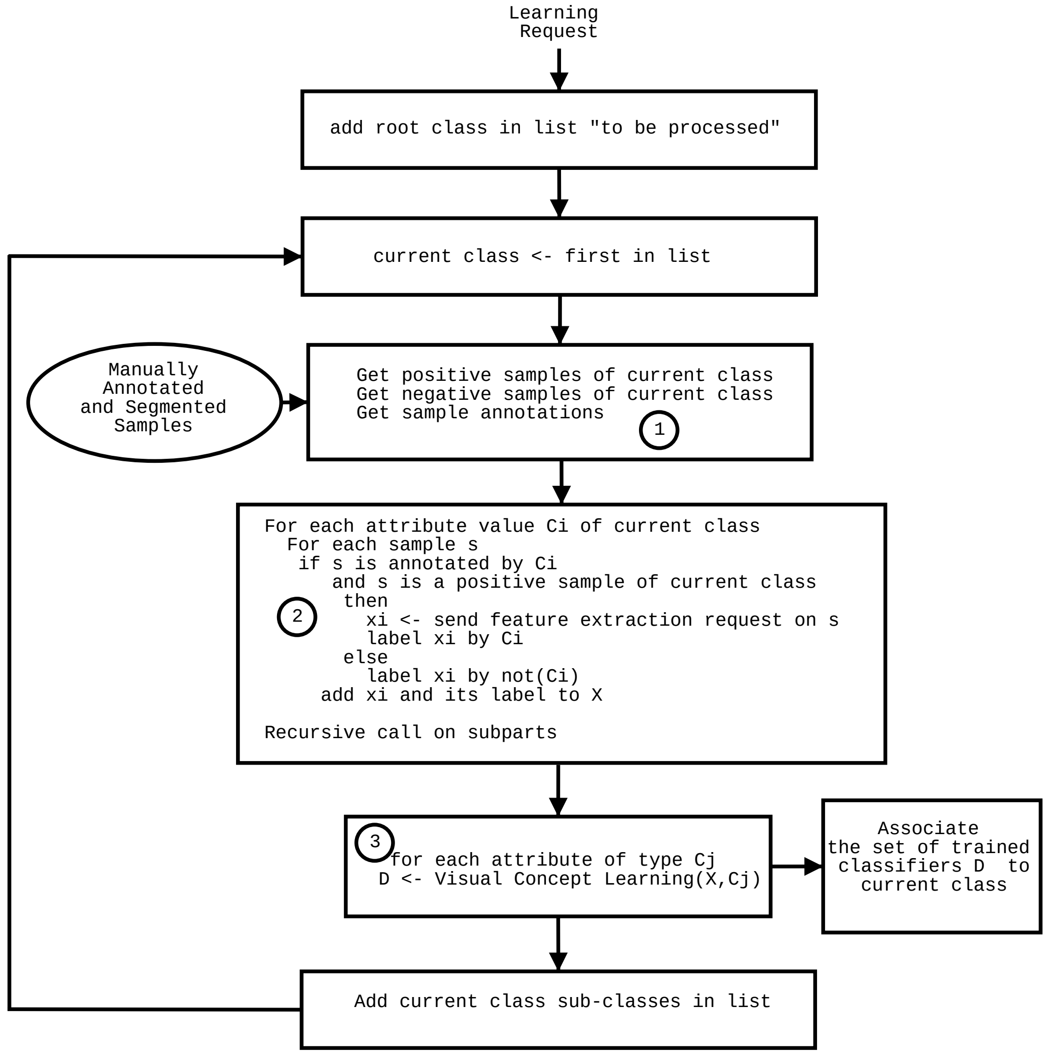

Fig. 13. Simplified version of object learning algorithm

during local matching.

(5) If object matched current class, the classification algorithm tries to go deeper in the domain class hierarchy. If matching fails, current class is dropped.

8 Discussion

The proposed approach allows semantic and explicit object categorization. The global architecture does not act as a black box and is able to explain categorization results.

One strong point is the modularity of the approach. New algorithms can be integrated in the segmentation module and in the feature extraction module.

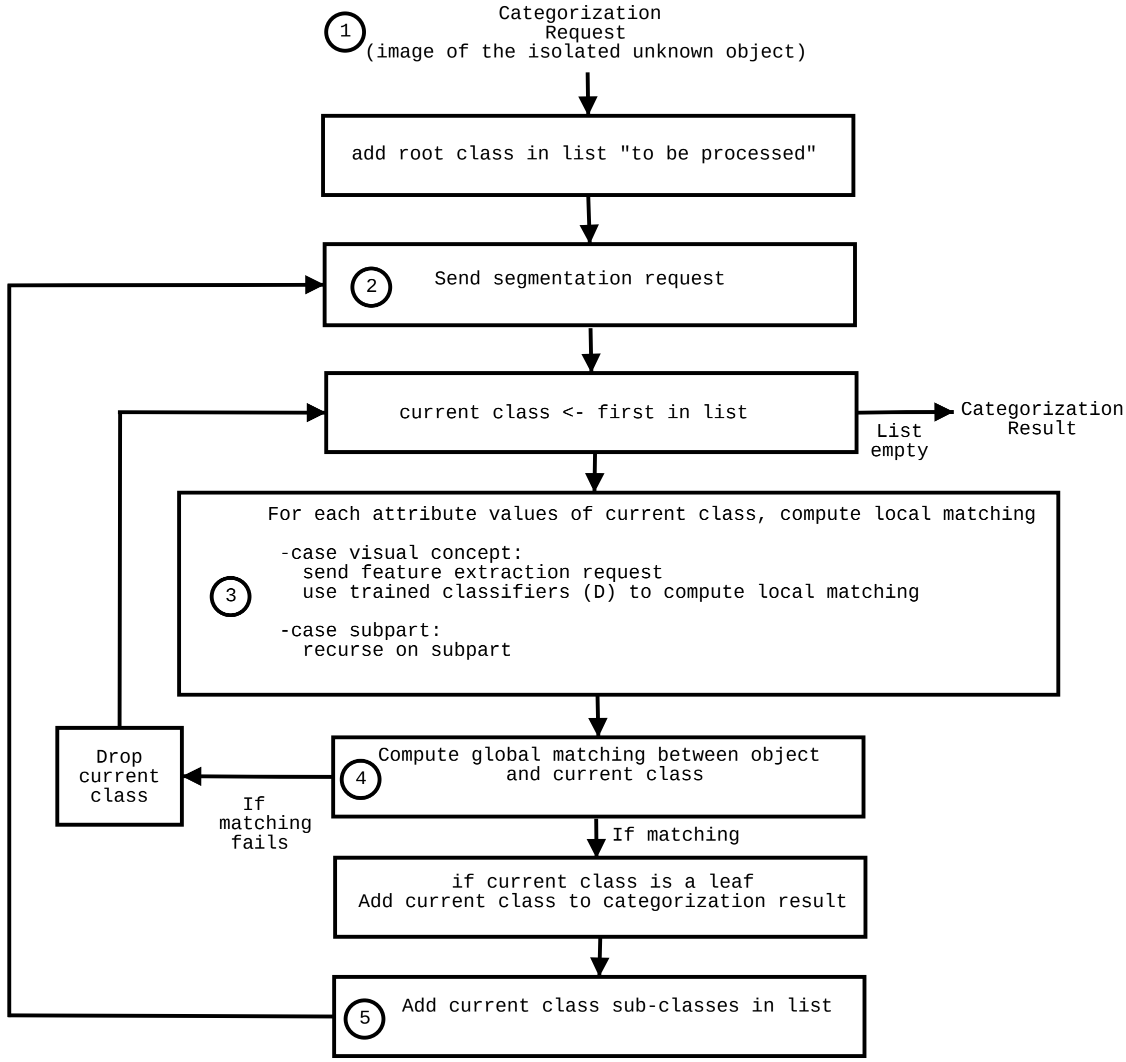

Fig. 14. Simplified Version of object categorization algorithm

The same goes for the visual concept learning module which currently uses k-nn, multi layer perceptrons and support vector machines. Changes in the low-level part of the architecture has no consequence on the high-level part.

One of the main advantages of our approach is that the visual concept ontology acts as a user-friendly intermediate between the image processing layer and the expert. Another interesting aspect is that the ontology can be used to describe objects of different domain of expertise. It is up to the learning layer to ground symbolic concepts in a different way.

At the segmentation level, a major remaining challenge is to define precisely the feedback to the segmentation level when object categorization fails. We are also planning to use program supervision techniques (see (17)) combined with learning techniques to improve segmentation quality. We have explained that visual concepts are included in segmentation requests. Trained classifiers associated with these visual concepts should be used to validate segmentation.

The proposed approach leans on a knowledge acquisition phase which has to be as complete as possible in order to make the learning phase and the categorization phase efficient. This raises the question of the balance that has to be found between knowledge which has to be provided by the expert and knowlege deduced from images samples. In the current implementation (see (11)), the expert has to provide an important amount of knowledge. An interesting future direction is the use of unsupervised learning techniques which could guide the expert by automatically filling class attributes. Indeed, similarity between samples could be used to infer similiar attribute values between classes.

9 Conclusion

This paper presents an original approach to complex object recognition. We mix the explicit aspect of knowledge based approaches with machine learning techniques which allow mapping between symbols and pixels. This approach is structured in three main phases. A knowledge acquisition phase which consists in describing a set of domain classes with visual concepts provided by a visual concept ontology. This ontology is composed the following types of visual concepts : spatial concepts and relations, color concept and texture concepts. The result is a domain knowledge base.

An object learning phase follows the knowledge acquisition process in order to obtain a knowledge base augmented by a set of classifiers trained to the recognition of the visual concepts used for the description of each classes.

The categorization phase tries to match an unknown object with one or several domain classes. The matching is done between visual concepts computed on the unknown object and visual concepts used for the description of domain classes.

There are several remaining issues. A good object segmentation as well as good subpart segmentations are needed in this approach. For some specific applications, this hypothesis is reasonable. In general, segmentation remains a major issue. We plan to use visual concepts, program supervision and learning techniques to deal with this problem. The visual concept ontology provides an efficient guide for knowledge acquisition but unsupervised learning techniques could make this task easier.

References

[1] C. Hudelot, M. Thonnat, A cognitive vision platform for automatic recognition of natural complex objects, in: International Conference on Tools with Artificial Intelligence, Sacramento, USA, 2003.

[2] M. Thonnat, A. Bijaoui., Knowledge-based galaxy classification systems, Knowledge-based systems in astronomy, Lecture Notes in Physics. 329, 1989.

[3] S. Liu, M. Thonnat, M. Berthod, Automatic classification of planktonic foraminifera by a knowledge-based system, in: The Tenth Conference on Artificial Intelligence for Applications, IEEE Computer Society Press, San Antonio, Texas, 1994, pp. 358–364.

[4] F. Xu, The role of language in acquiring object kind concepts in infancy, Cognition 85 (3) (2002) 223–250.

[5] G. Miller, P. Johnson-Laird, Language and Perception, Cambridge University Press, 1976.

[6] K. Mardia, Shape in images, in: S. Pal, A. Pal (Eds.), Pattern Recognition – From Classical to Modern Approaches, World Scientific, 2002, pp. 147–167.

[7] D. G. Lowe, Object Recognition From Local Scale-Invariant Features, in: International Conference on Computer Vision (ICCV), 1999, pp. 1150–1157.

[8] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: European Conference on Computer Vision, Springer, 2002, pp. 128–142, copenhagen.

[9] G. Dorkó, C. Schmid, Selection of scale invariant neighborhoods for object class recognition, in: Proceedings of the 9th International Conference on Computer Vision, Nice, France, 2003.

[10] T. Matsuyama, V.-S. Hwang, SIGMA - A Knowledge-Based Aerial Image Understanding System, Plenum Press New York USA, 1990.

[11] N. Maillot, M. Thonnat, A. Boucher, Towards ontology based cognitive vision, in: J. L. Crowley, J. H. Piater, M. Vincze, L. Paletta (Eds.), Computer Vision Systems, Third International Conference, ICVS, Vol. 2626 of Lecture Notes in Computer Science, Springer, 2003.

[12] S. Harnad, The symbol grounding problem, Physica D (42) (1990) 335–346.

[13] T. R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing, in: N. Guarino, R. Poli (Eds.), Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, Deventer, The Netherlands, 1993.

[14] A. Rao, G. Lohse, Towards a texture naming system: Identifying relevent dimensions of texture, Visual Research 36 (11) (1993) 1649–1669.

[15] E. Sciascio, F. M.Donini, M. Mongiello., Structured knowledge representation for image retrieval, Journal of Artificial Intelligence Research 16 (2002) 209–257.

[16] A. G. Cohn, S. M. Hazarika, Qualitative spatial representation and reasoning: An overview, Fundamenta Informaticae 46 (1-2) (2001) 1–29.

[17] C. Shekhar, S. Moisan, R. Vincent, P. Burlina, R. Chellappa, Knowledge-based control of vision systems, Image and Vision Computing 17 667–683.

[18] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: ACM Multimedia, 1996, pp. 65–73.

[19] B. Dubuisson, M. Masson, A statistical decision rule with incomplete knowledge about classes, Pattern Recognition 26 (1) (1993) 155–165.

[20] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature-selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.

[21] E. Choi, C. Lee, Feature extraction based on the bhattacharyya distance, Pattern Recognition 36 (8) (2003) 1703–1709.