Towards Domain Ontology Creation Based on a Taxonomy Structure in Computer Vision

Mansouri fatimaezzahra
Computer Science Engineering Laboratory
Department of Computer science
Marrakesh, Morocco

Elfazziki abdelaziz
Computer Science Engineering Laboratory
Department of Computer science
Marrakesh, Morocco

Sadgal mohamed
Computer Science Engineering Laboratory
Department of Computer science
Marrakesh, Morocco

Benchikhi loubna
Computer Science Engineering Laboratory
Department of Computer science
Marrakesh, Morocco

***Abstract*—In computer vision to create a knowledge base usable by information systems, we need a data structure facilitating the information access. Artificial intelligence community uses the ontologies to structure and represent the domain knowledge. This information structure can be used as a database of many geographic information systems (GIS) or information systems treating real objects for example road scenes, besides it can be utilized by other systems. For this, we provide a process to create a taxonomy structure based on new hierarchical image clustering method. The hierarchical relation is based on visual object features and contributes to build domain ontology.**

Keywords—Domain Ontology; Categorization; Taxonomy; Road scenes; Computer vision

I. Introduction

In this paper, we treat modeling problems and representation of road scenes content using knowledge engineering methods. We seek to define and organize the knowledge of a field of study (road scenes our field of study), through ontologies, which will allow us to define the domain concepts and relations between them. The creation of a domain ontology will facilitate the task of object recognition, similarities search and it will facilitate the decision-making task.

The domain ontology is a common vocabulary for researchers who need to share information on a subject as concepts and the relations between them. It allows a knowledge formal representation of a specific scientific field.

The knowledge organization by classes minimizes the information complexity and improves the efficiency of information processing. This process also allows new elements classification, besides the information subsequent use in decision-making, evaluative judgments, selection and generation of new knowledge.

We adopt a structuring method through image classification using a measuring function based on the cue-validity of attributes [1], which evaluates each partition and preserves the best according to certain criteria. Our function allows us to generate a tree structure (taxonomic tree) with several different levels from each other by their level of abstraction and precision. This classification as a taxonomic tree will be the basis of our ontology of hierarchy.

The objective of this research is to propose a methodology for automatic generation of a taxonomic tree as a basis for visual objects ontology building. This generation uses an assessment that can select each level of the tree in accordance with the criteria of the categories accuracy and the recognition time. It is based on the characterization of objects by the visual attributes and organizing them hierarchically by techniques of non-supervised learning.

This work is about the object recognition in cognitive vision. In the knowledge acquisition phase of the domain of study using a class hierarchy of objects and subclasses.

Each class will be described in terms of visual concepts (shape, color, texture) provided by an ontology. Each visual concept of this ontology is associated with descriptors, the semantic gap is reduced to the expert who will intervene to add relations between concepts and place the objects in their membership classes.

All this process will facilitate the decision making in practice. For road scenes for example it will help driving by detecting the obstacles in real time.

Section 2 of this article presents the state of art we present related works, followed by section 3, which evoke the ontology building methods, section 4 details our approach to create an automatic domain ontology based on a taxonomy structure. We present an example in section 5, then a conclusion to close the article.

II. State of the Art

Ontology is the study of the knowledge about the world. We define ontology also as the study of the organization and the nature of the world, regardless of their perception [2]. Sowa suggests that the subject of ontology is the study of things categories that exist or may exist in a certain area [3]. With the emergence of the knowledge engineering, the community introduces ontology in artificial Intelligence as a response to the problems of knowledge representation and manipulation within computer systems. The most common definition of ontologies is the Gruber definition [4]; he defines ontology as explicit formal specifications of the terms of a domain and relations between them.

In image processing as in other areas, we use ontologies for knowledge structuring. Some studies use ontologies from the stage of images segmentation. In [5] ontologies include parameters for the segmentation algorithm and the potential label areas. In visual ontology, the description of concepts is mainly based on the geometric characteristics. After initial segmentation, we adjust the segments to get closer to their description in the ontology.

In [6] Hindle presents an early work on automated taxonomy building in which the names are grouped into classes. Hearst seminal work on the use of linguistic models also aimed to discover the taxonomic relations [7]. Recently, Reinberger and Spyns [8] present an application of clustering techniques in the biomedical field. In [9] we find a view of all clustering approaches for ontological learning structures.

Vision systems based on knowledge have proven to be effective for complex object recognition and scene interpretation. They offer the possibility of reuse and extensibility. Furthermore, in knowledge-based systems, we separate domain knowledge from the image processing knowledge. This implies better traceability of the different sub-problems. In literature, we find a variety of statistical approaches based on machine learning for annotating automatically image regions: SVM, decision trees, artificial neural networks, Bayesian networks. These approaches [10] learn matching functions between the characteristics and the regions classes. Although describing well the visual image's content, statistical methods do not adequately represent the picture's meaning as perceived by humans because semantic is limited to the learning results of the function linking low-level features to high-level concepts. These performances also depend on the number of classes learned.

Besides the statistical methods, some works [11] propose to use the domain concepts to annotate images: free annotation where no vocabulary is predefined, annotation by key-words in a set of words (or concepts) is proposed to the user and annotation by ontology where a set of words and the relations between them are provided to the user. Using ontology aims to different goals: unified description of image characteristics, visual description of the relations between characteristics (lines, region...), use of contextual information and finally the reconciliation between visual and semantic level. In the purpose of knowledge formalization, Neumann [12] proposed to model the scenes using a logical description. The main contribution of logical descriptions is to avoid mistakes when modeling knowledge and inference are intuitively constructed. Clouard [13] proposed an image processing ontology example. It contains 279 concepts, 42 roles and restrictions 192. However, the domain knowledge is not present, and image context and the user's knowledge are not taken into account. These approaches and ontologies have limitations, especially concerning their reuse.

For ontology construction, only a few automatic methods are proposed [14, 15, 16, 17]. Elliman [16] propose a method for ontologies construction to represent a set of web pages on a specified website. We use the map organization to build the hierarchy. In our case, we automatically modify the tree and the label organization in the hierarchy nodes. Bodner proposes a construction method based on a statistical hierarchy [14]. In [15] Hoothe offer various clustering techniques to illustrate the text using ontologies. All hierarchies will be constructed for multiple viewing only not in the ontology construction purposes. In addition, all these ontology construction methods are used in the text field; however, we address this problem in the *image domain.*

Regarding image processing, Latifur Khan in [17] proposes a method of ontology automatic construction from the automatic classification algorithm with a similarity based on color and shape. The results lead to a precision measurement on 6 categories known in advance.

III. Ontology Structuring and Building

A. Definitions

An ontology is an explicit formal description of concepts (also called Classes) in a given field, properties of each concept describing attributes and the attribute restrictions. An ontology and all the class's individual instances constitute a knowledge base. There is actually a fine line between the end of an ontology and the beginning of a knowledge base. How to construct an ontology is still subject to much discussion in the community. Our understanding of the different contributions made so far is that there is a three distinguish construction options:

- *Bottom-up approach*: The ontology is constructed by generalization starting from the low taxonomic concepts layers. This approach encourages the creation of specific and adapted ontologies.
- *Down approach*: The ontology is built by starting with specialization in high taxonomic concepts layers. This approach encourages the reuse of ontologies.
- *Centrifugal approach*: Priority is given to the identification of the central concepts in the application that will be generalized and specialized to complete ontology. This approach encourages the emergence of thematic domains in the ontology and promotes modularity.

To formally present an ontology, we should give concepts a lexicon and explicit the relations between them.

According to Gruber, M. Uschold [18]:

- An ontology involves or includes a certain view of the world for a given domain. This view is conceived as a *set of concepts*, their *definitions* and their *inter-relations*. This is called a *conceptualization*.
- An ontology can take different forms, but it necessarily includes a term vocabulary and specification of their meaning.

- An ontology is a specification, making partial account of a conceptualization.

B. Formal structure of an ontology

The quintuplet O= {C, R, H^C, rel, A^O} presented by Steffen Staab in [19] is the ontology structure:

- C and R are disjoint sets of concepts and relations
- H^C Concepts hierarchy (taxonomy): $H^C \subseteq$ C x C, $H^C(C_1, C_2)$ means that C_1 is a sub-concept of C_2 (oriented relation)
- Rel: the relation rel: R $\rightarrow$ C x C (define semantic relation) with 2 associated functions:

1. dom: R $\rightarrow$ C with dom(R):= $\prod$1(rel(R))
2. range: R $\rightarrow$ C with range(R):= $\prod$2(rel(R)) co-domain
3. rel(R) = (C_1,C_2) is written like $R(C_1,C_2)$

Often it defines abstract ontology.

We can add a lexicon to the ontology O:= {C, R, H^C, rel, A^O} which is a quadruplet L:= {L^C, L^R, F, G}

- L^C and L^R: disjoint sets of lexical concepts and relations
- F, G: 2 relations called references F (L^C(for the concepts), G (L^R x R (for the relations), fork L $\in L^C$: F (L) = {C (C / (L, C) (F}, F^{-1}(L) = {L (L / (L, C) (F}
- Idem For G and G^{-1}

Then we obtain the concrete ontology: couple (O, L)

In ontologies, inter-relations connect all the concepts. If there is an inter-relations R, between concepts Ci and Cj, then there is also an inter-relations R′ between concepts Cj and Ci. In Figure 1, we represent inter-relations by labeled arcs/links. We used three kinds of inter-relations to create the ontology: Is-a, Instance-of, and Part-of. These correspond to key abstraction primitives in object-based and semantic data models [20].

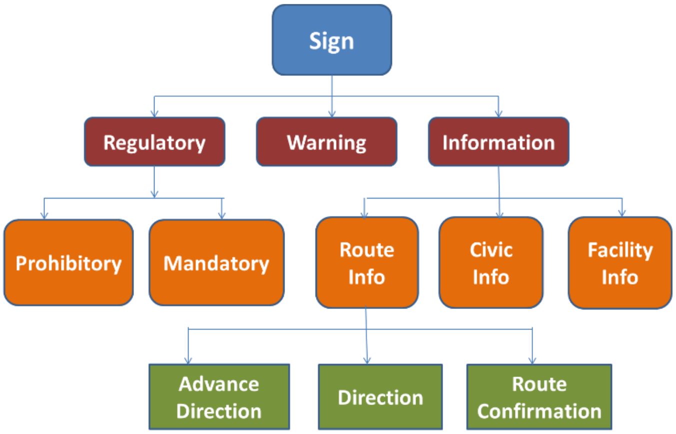


Fig. 1. Example of a road panels' tree

Figure 1 illustrates an example of an ontology for the road scenes domain. We obtain this ontology from generic terminology or from experts. The directed acyclic graph (DAG) describes it. Each node in the DAG represents a concept. The concepts in the ontology contain a label name and feature vector. A feature vector is a set of features and their weights. Each function can represent an object in an image.

Our contribution is to work on the concepts hierarchy, thus the automatic generation of H^c from clustering techniques on the objects aspects.

C. General ontology construction scheme

In a taxonomy, we organize the controlled vocabulary in a simple hierarchical format. This hierarchy often corresponds to a specialization. There is therefore a defined link between a term's vocabulary and its children. This link gives an extra meaning. From a controlled vocabulary, we can go to an organized vocabulary. In fact, even a lexicon or a taxonomy are forms of unformulated grammar ontology. When establishing a category and a hierarchy of this categorization, we establish dependencies between these terms. These hierarchies have meaning outside the vocabulary itself. An ontology corresponds to a controlled and organized vocabulary and the explicit formalization of relations created between the various terms of vocabulary. We can consider that the taxonomy is a semi-formal representation of the world while ontology gives a formal representation, the same representation will give us an operational model.

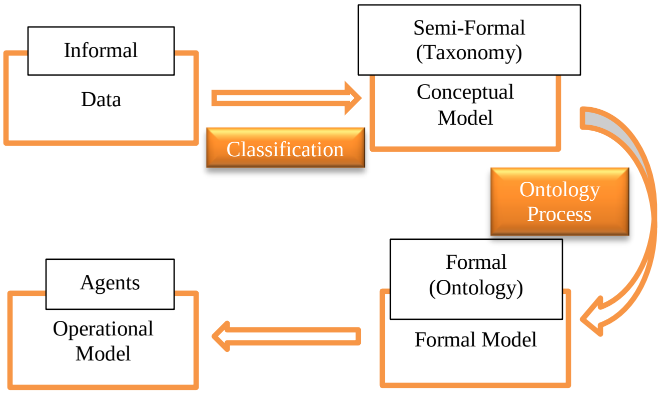


Fig. 2. From informal to formal model

IV. Proposed Approach

A. General architecture of the proposed system

Images number and resolution are increasing. We have then a tremendous increasing amount of information available but not exploitable. It is therefore necessary to develop semi-automatic processes, facilitating the organization of visual objects to minimize the recognition time while keeping the accuracy. In our ontology construction process, we adopt a down approach going from the top of the taxonomic tree.

Our goal is to present, formally, the concepts of the road scene domain using the ontology concepts to present knowledge. For this, we will be using the result of partitioning objects (taxonomic tree) while keeping the most consistent partitions giving the best recognition.

Our approach progress as the following figure:

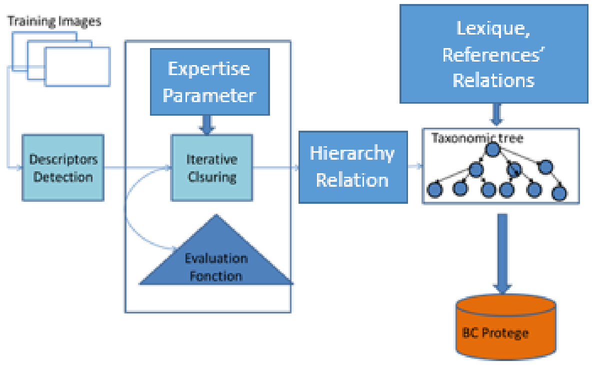


Fig. 3. General system architecture

It begins first with a visual features detection step of all images, then we go to the clustering step using an iterative clustering algorithm while evaluating each partition obtained, which will allow us to create a hierarchy that produces a taxonomic tree. After that, we create an ontology by adding a lexicon and relations between the taxonomy nodes. This ontology will be our knowledge base of the domain.

The following sections IV-B, C, D, E, F details all the steps.

B. Descriptors' selection

In this step, we choose descriptors that reflect the color, shape and texture of objects. We extract descriptors and subsequently store them in the database.

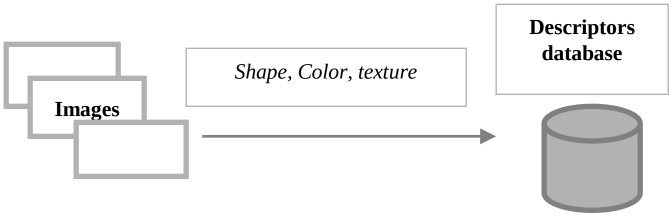


Fig. 4. Visual descriptor extraction

A Descriptor (feature) is a metric or any quantifiable value used to describe an image at a high-level perspective. Features related to color, texture, shapes, color blobs, corners are contained in an image. Color is a basic feature for image representation, and is invariant with respect to scaling, translation and rotation of an image [21]. It can be computed as Histograms (distribution of RGB, Hue,…), as SIFT descriptors or as moments of order p+q on RGB triplet or as a SIFT descriptor [22].

The haralick texture features are used for image classification. They capture information that emerges in patterns of texture. These kind of features is calculated by using co-occurrence matrix, which is computationally expensive [23].

In imaging applications, the shape of image objects provides a useful hint for similarity matching. For image retrieval the shape descriptor wants to be invariant to scaling, rotation and translation [24]. The descriptors are based on contour and region extraction.

In practice, we can combine multiple descriptors and choose those that are most significant (e.g. In some cases, we can neglect the texture if the objects have the same texture).

C. The Clustering method adopted

We use an unsupervised approach known as clustering. The main difference with the supervised classification is that the data set, from which we learn decision rules, does not include the information of an observation belonging to classes.

The problem of an automatic classification is to produce the labels {z1, ..., zn} of observations{x1, ..., xn} ∈ R*p* only on the knowledge of the values taken by the *p* variables. Unlike discriminant analysis, automatic classification does not have a learning phase to learn the classes' characteristics. An additional difficulty in automatic classification is that we do not necessarily know the number k of groups.

As in discriminant analysis, we divide clustering methods into two categories generative and discriminative methods. Generative clustering methods are quasi-exclusive, based on the mixture model and the EM estimation algorithm. Discriminative methods, in turn, all use a hierarchical classification structure.

We adopted a partitioning clustering method including K-means algorithm. By getting more partitions with K-means we produce a taxonomic tree.

D. Expert parameter and evaluation function

Our method of creating partitions does not aim to provide a hierarchical clustering algorithm, but to obtain a dynamic taxonomy. It is a structure for storing information in a memorable and operative way.

We construct the tree in an incremental way. At each level, we establish a partition of objects, meeting the criteria of accuracy and easy recognition. We need to have a measure to evaluate the partitions and retain the one with the mentioned criteria.

As the objects regrouping or separation depends on their looks or dissimilarity. We use the entropy measure [25] to establish partitions evaluation function. The process involves subsequently producing partitions and evaluating them. The difficulty in production methods is their efficiency to produce the most relevant partitions given the appearance NP-complexity.

It will add additional criteria to make combinations. Similarly, the user's interventions grow the system to introduce unnecessary hierarchy levels. Sometimes we get degraded situations where a node has only one son. There is therefore two simplification access to explore for a more refined structure.

We base the evaluation of a categorization on:

- The Cue-validity, which is the probability p(Cj | ai) of a category Cj having the descriptor ai.

- Category $p(ai \mid Cj)$ the probability of the existence of an attribute ai in a category Cj.

We have proposed in [1] a measure noted *Ep(Γ)* for a partition *Γ* in the taxonomic tree. It is based on cue-validity, which expresses the non-uniformity of properties in categories. Minimizing *Ep(Γ)* consist on building the highest possible homogeneous categories. This measure decreases every time we go down in the taxonomic tree.

On the other hand, recognition must be fast. It is therefore necessary to climb up the taxonomic tree before the matching process. Higher you go, the more the uncertainty of finding a category for an object decreases the top of the tree contains all objects. We have noted *Ec(Γ)* the measure that expresses the uncertainty of categories.

For that we can say, that to allow the operation of recognition to succeed in the majority of cases on a quick and robust way, we must find a partition that achieves the minimum of *Ep(Γ)* and *Ec(Γ)*.

Then we expressed this combination in a linear way:

$$E(\Gamma) = \alpha.Ep(\Gamma) + \beta.Ec(\Gamma) \quad (1) \quad [1].$$

Ec and *Ep* plays antagonist roles. The tree root has only one category, the effort is focused on attributes, in this case, we keep only Ep. Otherwise, the level of the sheets (each object is one category), The effort is focused on the categories: we retain only Ec. For this we choose $\beta = 1-\alpha$. $0 \leq \alpha \leq 1$, and we have: $E(\Gamma) = \alpha.Ep(\Gamma) + (1-\alpha)Ec(\Gamma)$.

By adjusting the parameter *α*, we change the partition level in the taxonomic tree that minimize E(Γ). In fact, this parameter expresses the degree of expertise in the object domain. The more knowledge we have, the closer we get to the leaves of the tree (α value approximates 1).

We propose many categorization procedures in an incremental way through minimizing the overall uncertainty *E(Γ)* in each step. Our method idea is to find the optimal partition in the sense of minimizing the total uncertainty E, by progressive construction of the partition. We start from a small town of a few objects made necessary for the calculation of uncertainty. We proceed by adding objects to an existing agglomeration and decide whether to merge or divide them. The problem may arise at the dividing level of possible combinations based on the number of content objects. The category search of the object is done by minimizing the overall uncertainty *E(Γ)*.

The α variation permits the construction of several levels, including the root level ($\alpha = 0$) and the leaf level ($\alpha = 1$). We will then have a taxonomic tree.

To build taxonomic tree allowing us to generate the domain ontology, we must seek to have a unique connection between the upper and the lower partition, in the sense that a lower class has an inclusion link to a single partition larger and not 2 or more.

E. *Taxonomic tree creation by hierarchical relations*

1) *Partitioning*

An ontology should:

- Enable the reuse of a domain knowledge.
- Explain what is considered implicit in a domain underlying an implementation, which makes the modification of specifications possible in the case of the evolution of the domain knowledge.
- Distinguish domain knowledge of operational knowledge.
- Analyze knowledge in a domain since formal analysis of terms is extremely valuable as well when you want to use existing ontologies, as when we want to extend those.

There is not a single methodology to create an ontology. On this paper, we will focus on the automatic generation of domain ontologies hierarchically using a top-down method.

In a previous article, we proposed a new method of categorization of visual objects. The proposed function is based on entropy (as we have explained above). It allows us to create a taxonomic tree of objects from different partitions. That is to say by changing the value of a variable α we get to generate partitions with a number of classes. The gathering of these partitions creates a taxonomic tree with several levels, the highest level brings all classes while the lowest level is composed of several classes with a higher accuracy.

Choosing a partition is done by maximizing homogeneity and promoting the recognition process. We explicit our approach as follows, in the first step we categorize objects using a clustering algorithm. We use our evaluation function to create a taxonomic tree and this by changing the α value that varies in the range [0,1].

Getting close to 1 gives us more classes (clusters) thus better accuracy during the step of recognition but recognition time becomes expensive.

The principle of the procedure is as follows:

```
For $\alpha$=0 to 1 by step of st do
  For nbclasse= nbClassMin To nbClassMax by a step of 1 Do
     -Apply the clustering Algorithm
     -Evaluate the score produced by our evaluation function (cost)
  End For
    -Retain the partition minimizing function for $\alpha$
End For
```

Note that the classes with the minimum cost increases with α. The algorithm then divides the current partition (more classes) when α increases.

α Value increasing

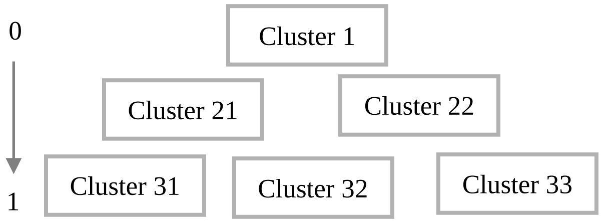


Fig. 5. Example of objects partitioning

2) Hierarchy process

As we mentioned α varies in the range [0,1], when it gets closer to 1 we get more classes per partition so we get a better accuracy.

The selected partition for each α value is the one with the minimum uncertainty E.

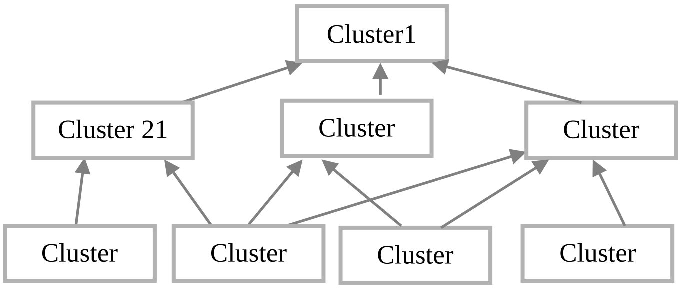


Fig. 6. Objects partitioning (decision step)

Cluster 1 is the first partition Cluster 21, 22 and 23 the second partition and the third partition is the cluster 31, 32, 33 and 34.

We can see that Cluster 32 objects belong to 3 clusters Cluster 21, Cluster 22 and Cluster 23.

The hierarchy property is an agglomeration of clusters in a low-level to form large clusters in a high-level. In order to decide which clusters should be combined (for agglomerative), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric and a linkage criterion, which specifies the dissimilarity of sets as a function of the pairwise distances of the observations in the sets.

By using *d* as the chosen metric, some commonly used linkage criteria between two sets of observations *A* and *B* are:

-Maximum or complete-linkage clustering:

$$\max\{\, d(a,b) : a \epsilon A, b \epsilon B \,\}$$

-Minimum or single-linkage clustering

$$\min\{\, d(a,b) : a\epsilon A, b \epsilon B \,\}$$

-Mean or average linkage clustering, or UPGMA (the Unweighted Pair Group Method with Arithmetic Mean).

$$\frac{1}{|A||B|}\sum_{a\,\epsilon\,A}\sum_{b\,\epsilon\,B} d(a,b)$$

-Centroid linkage clustering, or UPGMC (Unweighted centroid clustering)

-Centroid linkage clustering, or UPGMC (Unweighted centroid clustering)

$||C_s - C_t||$ where Cs and C_t are the centroids of clusters *s* and *t*, respectively

-Minimum energy clustering

$$\frac{2}{nm}\sum_{i,j=1}^{n,m}||ai-bj||2 - \frac{1}{n^2}\sum_{i,j=1}^{n}||ai-aj||2 - \frac{1}{m^2}\sum_{i,j=1}^{m}||bi-bj||2$$

Other linkage criteria include:

- The sum of all intra-cluster variance.
- The decrease in variance for the cluster being merged (Ward's criterion).
- The probability that candidate clusters spawn from the same distribution function (V-linkage).
- The product of in-degree and out-degree on a k-nearest-neighbor graph (graph degree linkage).
- The increment of some cluster descriptor (i.e., a quantity defined for measuring the quality of a cluster) after merging two clusters.

Assume the chosen link is ξ , the hierarchical relation can be obtained by:

$H^C(C_j, C)$ with C = arg ($\min_{Ci}$ ξ(Cj, Ci)), C_i in high level

For example, in figure 6 the objects of cluster32 are distributed over cluster21, cluster22 and cluster23. One way to decide for the hierarchy is to consider the UPGMC (Centroid linkage clustering) link, then we obtain H^C(cluster32, C) where C realizing the minimum of ($||C_{32} - C_{21}||$, $||C_{32} - C_{22}||$ and $||C_{32} - C_{23}||$) with C_{ij} is the centroid of the corresponding cluster.

F. Towards concrete ontology

1) Automatic labeling

After creating the tree, each cluster gets automatically a label produced according to the level class and the position of the class at that level (according to a particular agreement). We define the labels L_{ij} $1<i<N$ with N number of the level and $1<j<n_i$ with n_i class number in the level i, classes are the concepts of ontology.

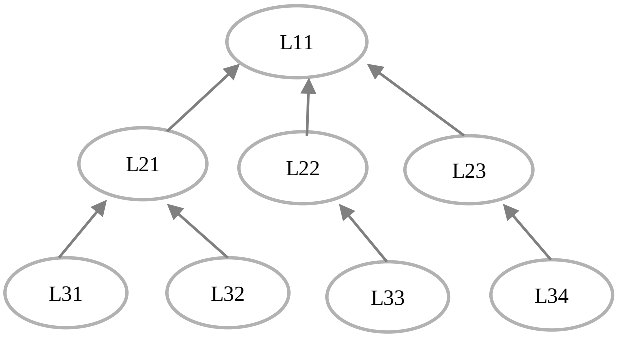


Fig. 7. Attributing labels to the ontology concepts

2) *Automatic generation of OWL code*

Then we will write our ontology in owl language and generate the graphic tree later using an ontology editor, respecting automatic labeling Ln.

```
refix name="untitled-ontology-3"
IRI="http://www.semanticweb.org/ontologies/2015/9/2/
untitled-ontology-3#"/>
    <Declaration>
        <Class IRI="#L11"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L21"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L22"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L23"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L31"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L32"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L33"/>
    </Declaration>
            <Declaration>
        <Class IRI="#L34"/>
    </Declaration>
       <SubClassOf>
        <Class IRI="#L21"/>
        <Class IRI="#L11"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#L23"/>
        <Class IRI="#L11"/>
    </SubClassOf>
```

Fig. 8. Writing ontology in OWL (Part of the code)

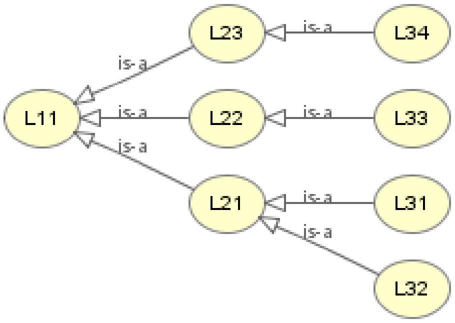
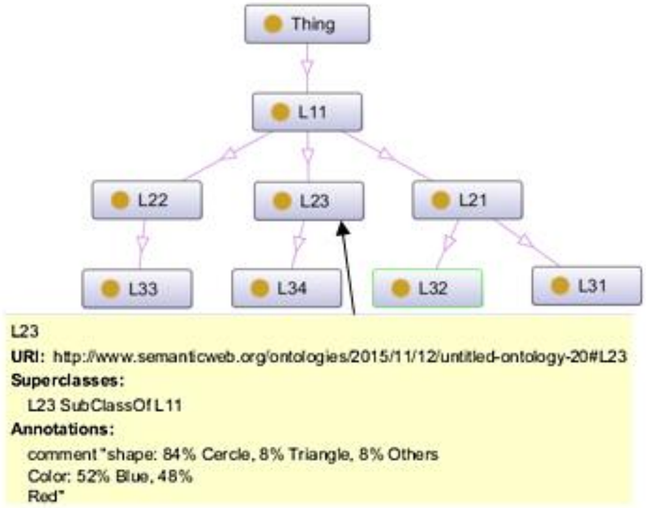


Fig. 9. Reading OWL file on Protégé and the generation of the tree

3) *The expert intervention*



L23
URI: http://www.semanticweb.org/ontologies/2015/11/12/untitled-ontology-20#L23
Superclasses:
L23 SubClassOf L11
Annotations:
comment "shape: 84% Cercle, 8% Triangle, 8% Others
Color: 52% Blue, 48%
Red"

Fig. 10. Generating the ontology graph from the OWL file with annotations

Now the domain expert can intervene to add other concepts to the ontology or to place new objects in their home concept by comparing their features with concept prototypes. From the taxonomic tree tagged with a detail on the homogeneity (shape, color, texture) of each class, a domain expert can intervene in this step to match the labels generated automatically by a lexicon belonging to the domain.

In the domain of the road scene, for example, and in particular for road signs, the expert can apply a decision tree similar to the figure:

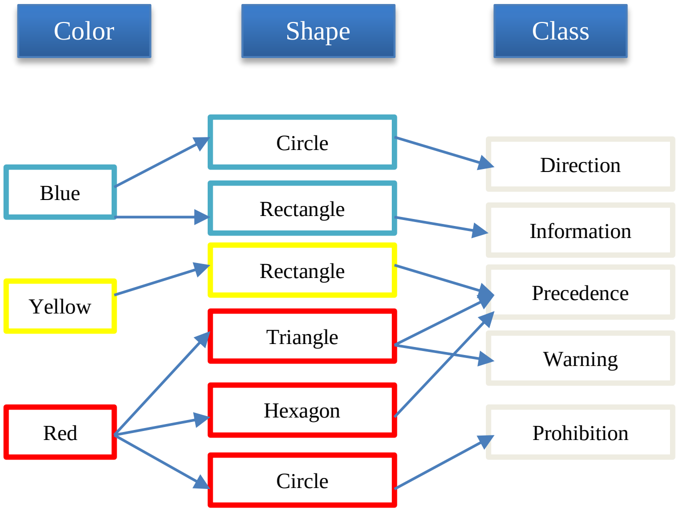


Fig. 11. Decision tree example for traffic signs

The expert can also adjust the tree with his domain knowledge (road scenes). The domain expert can intervene or not to keep the homogeneity of classes (concepts) and keep only the relevant tree.

V. Application Example

Goal: Creation of an ontology of hierarchy of road sign panels by implementing the different steps of our approach.

We considered 81 road signs images. The first step is to extract visual features (as shown above). We used color, shape and texture descriptors using 12 features.

TABLE I. Texture and Shape Descriptors

Descriptors	Definition
Entropy (F1)	Statistical measure of randomness. It characterizes the degree of organization
Contrast (F2)	Measures the local variation of the co-occurrence matrix of gray level
Correlation (F3)	Measures the joint probability of occurrence of specified pairs of pixels
Energy (F4)	Provides the squares sum of GLCM elements. Also known as uniformity or the angular second moment
Homogeneity (F5)	Measures the closeness of element distribution in the GLCM to the GLCM diagonal
Area (F6)	The number of pixels in the region
Perimeter (F7)	The perimeter of the distance between each adjacent pair of pixels around the region borders.
Convex area (F8)	The scalar indicating the number of pixels in Convex Image. This property is supported only for 2-D for input label matrices
Euler number (F9)	The scalar indicating the number of objects in the region, less the number of holes in objects.
Extent (F10)	The scalar that specifies the number of pixels in the pixel region of the total area delimitation. Calculated as divided by the area of the bounding box area. This property is supported only for 2-D input label matrices.

For color features we use the moments of second-degree features, calculating the energy derivatives, according to star and circle structure F11 F12.

TABLE II. A View of the Database of the 12 Features Used [1]

	F1	F2	F3	F4	F5	F6
Image 1	+3.154	+1.875	+0.398	+0.839	+0.879	+2148.000
Image 2	+3.275	+2.152	+0.344	+0.840	+0.866	+2157.000
Image 3	+2.910	+1.394	+0.446	+0.867	+0.899	+2166.000
Image 4	+3.367	+1.941	+0.337	+0.853	+0.864	+3250.000
Image 5	+3.163	+1.092	+0.397	+0.861	+0.895	+4.000
	F7	F8	F9	F10	F11	F12
Image 1	+403.647	+5104.000	-14.000	+0.244	+0.013	+0.005
Image 2	+313.421	+5109.000	-19.000	+0.245	+0.020	+0.006
Image 3	+313.421	+5110.000	-22.000	+0.246	+0.012	+0.006
Image 4	+313.421	+5104.000	-21.000	+0.369	+0.018	+0.009
Image 5	+6.000	+4.000	+1.000	+1.000	+0.010	+0.003

After extracting the features we classify the object using a clustering algorithm (kmeans) and use our evaluation function to keep the best partition for each iteration, we remind that the function is based on the entropy and by adjusting the α value we get different partitions (number of classes of each partition). In this application, we get a set of partition by varying α between 0,1 and 0,2 which gave us partitions with 2 to 8 classes as seen in table 3.

TABLE III. Partitioning Results Combining Color, Shape and Texture Descriptors Using Kmeans

	Color/Shape/Texture				
α value	0,1	0,12	0,15	0,17	0,2
Classes (number)	2	3	4	6	8

We consider this result and we create a hierarchy between those classes based on the centroid clustering or as we called it UPGMC (The Unweighted centroid clustering). UPGMC joins the objects or groups that have the highest similarity (or the closest distance), by replacing all the objects of the group produced by the centroid of the group. This centroid is considered as a single object at the next clustering step.

UPGMC, as well as WPGMC, can sometimes produce reversals in the dendrogram. This situation occurred in our example. This happens when:

- Two objects about to join (let us call them A and B) are closer to one another than each of them is with a third object C: ||AB||<||BC||
- After the fusion of A and B, the centroid of the new group A-B is closer to C than A was to B before the fusion: ||C_{AB}C||<||AB||

To make the decision of assigning object to the closest centroid, we need to measure the proximity. We followed the steps of proximity measurement explained in the hierarchy process section.

This gave us the following result (figure 12). For the first partition with two classes, the first one has mostly the prohibition and indication panels (with round shape), in the second class we get essentially danger panels (with triangular shape). Coming down on the tree we have partitions with more classes so we get more accuracy. In the last level of the 4 classes, we have the correct categorization. A class for prohibition panels, a class for intersection panels, a class for danger panels and a class for indication panels.

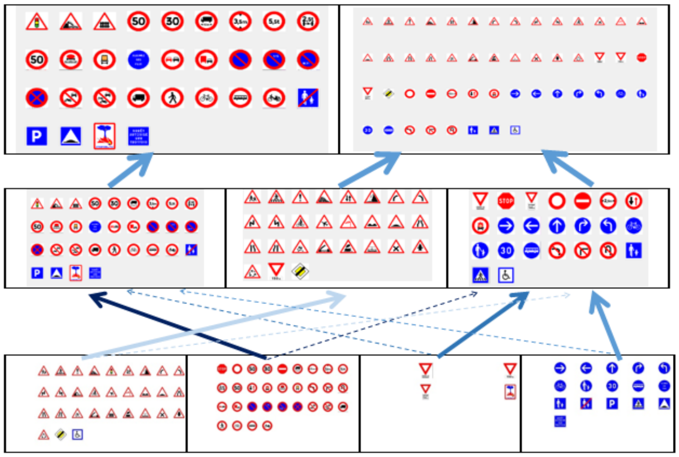


Fig. 12. Partitioning levels (Thick lines describe the final decision for the relation H^c with centroids distance, Thin lines describe short distances on centroids)

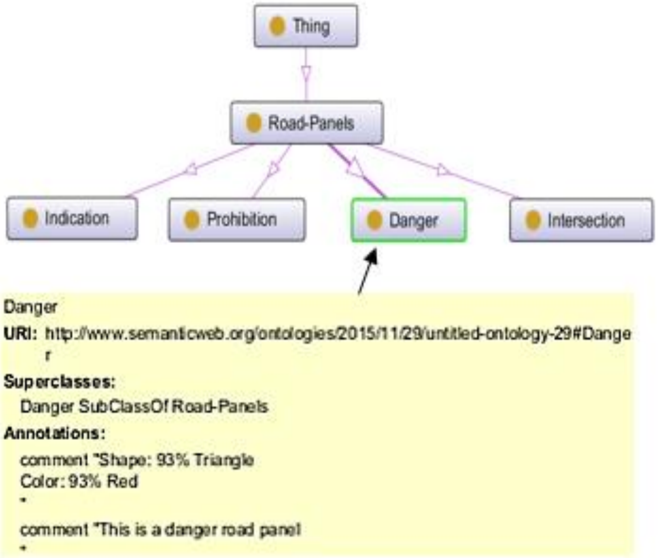
After having the classification results and the taxonomic tree now we generate our ontology of hierarchy by creating links between concepts. After that we write our OWL ontology, this file will be read later on an ontology editor where the expert can intervene.

```
<Declaration>
        <Class IRI="#Danger"/>
       <owl:AnnotationProperty
rdf:about="&dc;creator"/>
<owl:Class rdf:about="#Danger">
  <rdfs:label>this is the Class of Danger Road
Panels: Shape:93% Triangle, Color: 93%
Red</rdfs:label>
</owl:Class>
```

```
    </Declaration>
<Declaration>
        <Class IRI="#Indication"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Prohibition"/>
    </Declaration>
<Declaration>
        <Class IRI="#Intersection"/>
    </Declaration>
<SubClassOf>
        <Class IRI="#Danger"/>
        <Class IRI="#Road-Panels"/>
    </SubClassOf>
<SubClassOf>
        <Class IRI="#Indication"/>
        <Class IRI="#Road-Panels"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Prohibition"/>
        <Class IRI="#Road-Panels"/>
    </SubClassOf>
<SubClassOf>
        <Class IRI="#Intersection"/>
        <Class IRI="#Road-Panels"/>
    </SubClassOf>
```

Fig. 13. Writing the ontology in OWL (example panels) (part of the ontology)

This figure replaced the Lexicon by clusters' names to give them a meaning, but as mentioned above we give to each class a lexicon as Lij.



Danger
URI: http://www.semanticweb.org/ontologies/2015/11/29/untitled-ontology-29#Danger
Superclasses:
Danger SubClassOf Road-Panels
Annotations:
comment "Shape: 93% Triangle
Color: 93% Red
"
comment "This is a danger road panel
"

Fig. 14. Reading the OWL file in Protege (Danger Class with the annotations)

VI. Conclusion and Perspective

This paper presented an approach to object recognition and domain ontology creation from taxonomic tree. Our approach takes place in a set of steps, a knowledge acquisition step, where we describe a set of classes with visual concepts provided by a visual ontology concept.

We proposed in this article our current vision of domain ontologies we start from the image to identify the concepts and relations between concepts that emerge. In particular, we have shown that a methodology based on producing a consensus clustering (thus unsupervised) highlights semantically relevant concepts and relations.

We plan to develop this ontology as a basic vocabulary in a multi-agent system for road scene interpretation.

References

[1] Mansouri F, Sadgal M, Elfazziki A, Benchikhi L, An evaluation of perceptual classification led by cognitive models in traffic scenes, Electronic Letters on Computer Vision and Image Analysis, vol 14(1) pp 38-60 2015

[2] Nicola Guarino, Formal ontology, conceptual analysis and knowledge representation, International Journal of Human-Computer Studies, Volume 43, Issues 5–6, November 1995, pp 625–640

[3] J F Sowa, Ontology, Metadata, and Semiotics, Chapter Conceptual Structures: Logical, Linguistic, and Computational Issues,Volume 1867 of the series Lecture Notes in Computer Science pp 55-81

[4] Gruber T.R. 1995. Toward principles for the design of ontologies used for knowledge sharing.Int J Hum, Comput Stud, vol 43, 1995, pp 907-928.

[5] Hassan S, Hétroy F, Palombi O, segmentation de maillage guidée par une ontologie, 22 ème journée de l'association francophone de d'informatique graphique, Arles, 2009

[6] Donald HindleAT&T Bell Laboratories, Murray Hill, NJ Noun classification from predicate-argument structures, Proceeding ACL '90 Proceedings of the 28th annual meeting on Association for Computational Linguistics, pp 268-275

[7] Marti A. HearstUniversity of California, Berkeley, CA, Automatic acquisition of hyponyms from large text corpora Proceeding COLING '92 Proceedings of the 14th conference on Computational linguistics - Vol 2, pp : 539-545

[8] Peter Spyns, Marie-Laure Reinberger, Lexically Evaluating Ontology Triples Generated Automatically from Texts,Chapter The Semantic Web: Research and Applications, Volume 3532 of the series Lecture Notes in Computer Science pp: 563-577

[9] Philipp Cimiano, Andreas Hotho, Steffen Staab, Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis, Journal of Artificial Intelligence Research 24 (2005) 305–339.

[10] zhang G, Satya S, Samden D, From classification to epilepsy ontology and informatics, Epilepsia, Vol 53, Issue Supplement s2, pp: 28–32, July 2012

[11] Kompatsiaris Y., Hobson P. (ed.) 2008. Semantic Multimedia and Ontologies, vol. 1, Theory and applications springer

[12] Neumann B., Möller R., (2008). On scene interpretation with description logics. Image vision comput, vol 26, january, pp 82-101

[13] Clouard R., Renouf A., Marinette R. (2010). An Ontology-Based Model for representing image processing application objectives. International Journal of Pattern Recognition and Artificial Intelligence, vol 24, pp 1181-1208

[14] R. Bodner and F. Song, “Knowledge-based Approaches to Query Expansion in Information Retrieval,” in Proc. of Advances in Artificial Intelligence, pp. 146-158, New York, Springer.

[15] Dave Elliman, J. Rafael G. Pulido. “Automatic Derivation of On-line Document Ontology”, International Workshop on Mechanisms for Enterprise Integration: From Objects to Ontology (MERIT 2001) 15th European Conference on Object Oriented Programming, Budapest, Hungary, Jun 2001.

[16] A. Hotho, A. Mädche, A., S. Staab, “Ontology-based Text Clustering,” Workshop Text Learning: Beyond Supervision, 2001.

[17] Khan L, Wang L. Automatic ontology derivation using clustering for image classification. In: Eighth International Workshop on Multimedia Information Systems, Tempe, Arizona, November 2002

[18] MIKE USCHOLD, Knowledge level modelling: concepts and terminology The Knowledge Engineering Review The Knowledge Engineering Review / Volume 13 / Issue 01 / March 1998, pp 5-29

[19] Steffen Staab, Text Clustering Based on Good Aggregations Andreas Hotho, Alexander Maedche, Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA

[20] D. D. Dhobale, B. S. Patil, S. B. Patil, V. R. Ghorpade, Semantic understanding of Image content, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011,pp 191-195

[21] Automatic Image Annotation Using Group Sparsity. Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li, Dimitris N. Metaxas, Department of Computer Science, Rutgers,University, Piscataway, NJ, 08854 Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA

[22] Rekhil M Kumar, A Survey on Image Feature Descriptors, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7668-7673

[23] B. S. Manjunath, Member, IEEE, Jens-Rainer Ohm, Member, IEEE, Vinod V. Vasudevan, Member, IEEE, and Akio Yamada, Color and Texture Descriptors, ieee transactions on circuits and systems for video technology, vol. 11, no. 6, june 2001

[24] MPEG-7 Visual Shape Descriptors Miroslaw Bober ,ieee transactions on circuits and systems for video technology, vol. 11, no. 6, june 2001

[25] Shannon and Weaver. The mathematical theory of communication, 1949.