

# Semantic Identification: Balancing between Complexity and Validity

M. Falelakis, C. Diou, and A. Delopoulos

Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, GR 541 24, Thessaloniki, Greece

Received 1 September 2004; Revised 15 March 2005; Accepted 9 May 2005

An efficient scheme for identifying semantic entities within data sets such as multimedia documents, scenes, signals, and so forth, is proposed in this work. Expression of semantic entities in terms of syntactic properties is modelled with appropriately defined finite automata, which also model the identification procedure. Based on the structure and properties of these automata, formal definitions of attained validity and certainty and also required complexity are defined as metrics of identification efficiency. The main contribution of the paper relies on organizing the identification and search procedure in a way that maximizes its validity for bounded complexity budgets and reversely minimizes computational complexity for a given required validity threshold. The associated optimization problem is solved by using dynamic programming. Finally, a set of experiments provides insight to the introduced theoretical framework.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

## 1. INTRODUCTION

Semantic analysis of multimedia data tends to become a necessity in applications coping with the need to organize and/or understand digital content and provide high levels of interaction between humans and computers. It is against this background that researchers have recently put great effort on developing semantic extraction algorithms and frameworks for standardizing semantic descriptions, such as the MPEG-7. Moreover, interesting automatic annotation schemes for multimodal data sets have been proposed (see, e.g., [1]).

The procedure of semantic search/indexing is essentially equivalent to the computation of the *degree* that a semantic entity (e.g., an event, an object, a concept, etc.) is identified within a particular environment (e.g., a multimedia document [2–4], especially in the framework of MPEG-7 [5, 6], a scene in computer vision applications [7, 8], a set of multisensor measurements in the case of surveillance systems, etc.).

In the framework of this work, the identification procedure relies on a simple type of knowledge base, which contains formal definitions of all searchable semantic entities. In fact, a *hierarchical scheme* is adopted where each semantic entity is defined by decomposing it into either “simpler” semantic entities or elementary properties that can be quantified and are reserving the name *syntactic entities*. Very naturally, identification of a particular semantic entity decomposes to identify all or a subset of the syntactic entities that

either directly or indirectly participate to its definition. The involved computations and the resulting *identification degrees* correspond to fuzzy operations and membership values in our setup, respectively.

Fuzzy inference computations are assumed to obey the “modus ponens” approach, for example, a semantic entity  $A$  is decomposed to the (simpler) semantic and/or syntactic entities  $X, Y, Z$  in the sense that identification of any of  $X, Y, Z$  implies identification of  $A$  to a certain degree (relation value)  $F_{AX}, F_{AY}, F_{AZ} \in [0, 1]$ , respectively. The whole collection of (i) semantic entities definitions, (ii) algorithms employed to quantify syntactic entities, (iii) relation values, constitute what we call “semantic encyclopedia” (see also [9, 10] and references therein for similar definitions of semantic encyclopedias). The construction of a semantic encyclopedia as a means to bridge the so-called “semantic gap” is not a trivial task at all and is the subject of much ongoing research.

However, present work assumes the existence of such an encyclopedia and focuses on two main contributions regarding its use for semantic search and indexing. The first is the modelling of the use of the aforementioned “semantic” hierarchical schemes by means of *finite automata*. The second, and the most important, is the *design of efficient methods* for the computation of identification degrees taking into account the tradeoff between limitations of computational cost (i.e., algorithmic complexity) versus obtained validity of the identification.

A few comments are in order for the second target of the paper. Semantic identification is powerful and flexible due to its ability to collect and rank information and to make inference on the basis of a set—call it *scope* hereafter—of *versatile* measured features (corresponding to the aforementioned syntactic entities). The number of these features may grow exponentially with respect to the level of “semanticness.” If for example semantic entity  $A$  is defined on the basis of semantic entities  $B$  and  $C$ , the scope of  $A$  is equal to the union of the scopes of  $B$  and  $C$  and so forth. In addition, the richer and more detailed the semantic encyclopedia is, the larger the scope of each semantic entity becomes. For a given encyclopedia, the most valid answer regarding the identification or not of a certain semantic entity is achieved when all syntactic features included in its scope have been evaluated. Unfortunately, this may be impossible in practice; evaluation of each single syntactic feature consumes processing power (corresponding to the computational complexity of an associated algorithm) and the accumulated processing cost may exceed the available resources. This is particularly evident in real-time and/or bulky search/indexing procedures where hard limits on computational budget are inevitable (see [11]). Present work proposes optimization strategies that allow for selecting this subset of the scope that yields maximum validity under computational budget constraints. The dual problem is also solved, that is, that of selecting those syntactic features whose algorithmic evaluation yields some preset level of validity at the minimum computational cost. Formal expressions of complexity and validity are derived and the optimization problem is expressed as a variant of the so-called knapsack problem that is solved by using dynamic programming.

Section 2 is devoted to the formal definition of semantic and syntactic entities as elements of the semantic encyclopedia. Fuzzy sets approach is extensively used to describe the dependence of semantic entities on lower-level semantic or syntactic features. Decomposition of the definition of (complex) semantic entities to syntactic features alone is formalized within this section as well.

Modelling of semantic identification procedures by means of finite automata is introduced in Section 3. Finite automata operating on an alphabet consisting of the syntactic features (or equivalently the associated evaluation algorithms) describe the stages of a semantic search procedure. Within the same section, the aforementioned formal expressions for validity and complexity are introduced in conjunction with expressions for *certainty*, that is, the final outcome of the identification procedure which corresponds to the degree (a fuzzy quantity) up to which a particular semantic entity has been identified within the processed data.

In Section 4, we formally state the problem of finding the tradeoff between validity and complexity as a typical constrained optimization problem while in Section 5, we propose efficient algorithms for solving this optimization problem by means of dynamic programming.

A set of experiments is presented in Section 6 in order to illustrate the use of the proposed semantic encyclopedia and to provide insight regarding the proposed optimization

scheme. The first is a more or less didactic example that adopts a simple semantic definition of a “table drawing.” The next experiment tests the applicability of the proposed tools to the popular framework of sport-event annotation. The last one evaluates the proposed optimization scheme on large-sized definition scopes providing experimental statistics supporting the argument that by optimally selecting the subset of syntactic features to be evaluated may yield significant validity scores even for low complexity budgets. Finally, comments on the obtained results and a listing of open issues have been included in our last section.

## 2. FUZZY SEMANTIC ENCYCLOPEDIA

As stated before, our approach heavily relies on the existence of a semantic encyclopedia, which consists of *definitions* of high-level semantic entities in terms of other semantic entities, or lower-level measurable features (syntactic entities).

### 2.1. Syntactic entities

As *syntactic feature*  $t$ , we define any measurable quantity (e.g., brightness, frequency, straightness, etc.) that can be obtained by applying a *corresponding algorithm*  $\tau$  on the given data set (e.g., a scene, an image, a signal, etc.). For simplicity, we assume real-valued syntactic features, either 1-dimensional (e.g., brightness on  $\mathbb{R}$ ), or multidimensional (e.g., color on  $\mathbb{R}^3$ ).

A *syntactic entity* or property  $y_i(t) \in [0, 1]$  is a fuzzy set [12] on a syntactic feature  $t$ . For instance, the property “very bright” is defined on the feature “brightness” and the property “red” is defined on the feature color. We assign the label  $Y_i$  to a particular syntactic entity  $y_i(t)$  and assume a finite set  $\mathbf{Y} = \{Y_i\}$  of such labels corresponding to the entire collection of syntactic entities of interest.

It is essential to point out that the aforementioned computational cost of the search procedure refers to the algorithms  $\tau$  employed for measuring the data set under examination. Let  $t_\tau$  be a feature evaluated by employing the algorithm  $\tau$ , then  $\tau$  is used to assess the degree  $\mu_{Y_i} \equiv y_i(t_\tau)$  up to which the particular data set assumes property  $Y_i$ .

### 2.2. Semantic entities

The term *semantic entity* refers to higher-level objects or concepts that cannot be directly measured and are closer to human perception. We assign each semantic entity a label  $E_i \in \mathbf{E}$ , where  $\mathbf{E}$  is the set of all semantic entities considered.

The semantic encyclopedia is built on the assumption that each semantic entity can be described using other lower-level semantic as well as syntactic entities. We also assume that multiple descriptions of the same entity are allowed. Considering the descriptions depicted in Figure 1, entity  $A$  is described in two different ways: one in terms of  $a$  and  $d$  and another in terms of  $a$  and  $C$ .  $C$  is also a semantic entity and is in turn described by  $b$  and  $c$ . Notice that lowercase and uppercase letters denote syntactic and semantic entities, respectively.

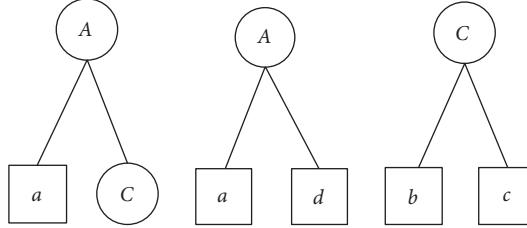


FIGURE 1: Simple descriptions.

Following a “modus ponens” approach, according to the first description, existence of  $a$  and  $b$  implies existence of  $\mathcal{A}$ . Thus the encyclopedia employs just the implication operator. This choice is not restrictive, however; depending on the needs of each application, one may choose to construct the encyclopedia using mathematical logics with more expressive power, such as description logics (see, e.g., [13, 14]).

### 2.3. Definitions

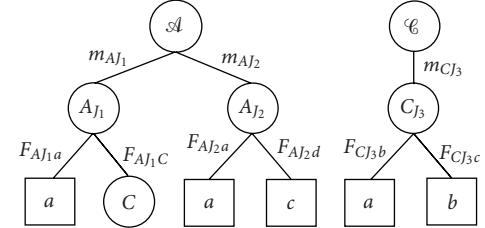
In most cases, existence of a set of entities ( $a$  and  $b$  in the example above) implies the existence of a semantic entity *up to a certain degree*, that is, we cannot provide strict descriptions of semantic entities, whose meaning is often objective or depends on the environment in which they are used. Hence the presented qualitative description of a semantic entity on the basis of simpler entities can be enriched by more quantitative information, regarding the degree of relation between a semantic entity and its successors, by considering the semantic entities as fuzzy sets whose members are the entities participating in the description.

Consider the set  $S \equiv Y \cup E$  of all semantic and syntactic entities. The importance of each entity participating in a description is quantified by an associated weight, which can be considered as an element of a fuzzy relation on  $S \times S$ . For a particular semantic entity  $E_k \in E$  in a description  $J$ , we define  $F_{kJ} : S - E_k \rightarrow [0, 1]$  for those  $S_i \in S$ , participating in  $J$ . We define as a *primary definition*  $E_{kJ}$  of  $E_k$  in terms of  $J$  the discrete fuzzy set

$$E_{kJ} = F_{kJ1}/S_1 + F_{kJ2}/S_2 + \dots + F_{kJn}/S_n. \quad (1)$$

Note that, for example, identification of  $S_1$  implies existence of  $E_{kJ}$  up to the degree  $F_{kJ1}$ .

Moreover, since there can be more than one alternative description for an entity, providing different amount of information about it, we define as *reliability*  $m_{kJ}$  of a description  $J$  of  $E_k$  a real number in  $[0, 1]$  measuring the quality and the amount of information provided by  $J$ . Equivalently,  $m_{kJ}$  is the degree up to which the particular description characterizes  $E_k$ . Figure 2 shows examples of primary definitions corresponding to the descriptions of Figure 1. Notice that entity  $\mathcal{A}$  has two alternative definitions  $A_{J_1}$  and  $A_{J_2}$ . Definitions of the form of (1) can be included in the encyclopedia, and are called *primary definitions*, or can be derived through a substitution procedure. A definition containing only syntactic entities is called a *detailed definition* and is of

FIGURE 2: Primary definitions: two alternative definitions for semantic entity  $\mathcal{A}$  and one for  $\mathcal{C}$ .

the form

$$E_{kJ_d} = F_{kJ_d1}/Y_1 + F_{kJ_d2}/Y_2 + \dots + F_{kJ_dm}/Y_m. \quad (2)$$

In our approach, every primary definition  $E_{kJ}$  can be transformed into a detailed one by replacing the semantic entities of  $J$  with their corresponding definitions and by repeating this procedure recursively. Resulting from the substitution procedure, a multiplicity of alternative detailed definitions for  $E_k$  may occur. This is due to the existence of more than one definitions for its successors  $S_i$ .

### 2.4. Generation of detailed definitions

Substitution of a semantic entity requires the assignment of new relevance factor values (weights) to the definition that occurs. Consider again the example of Figure 2. By substituting  $\mathcal{C}$  in the description  $J_1$ , we come up with a new description  $J_d$  of  $\mathcal{A}$  which depends on the syntactic entities  $a$  and  $b$  and the corresponding definition  $A_{J_d}$ ,

$$A_{J_d} = F_{AJ_da}/a + F_{AJ_db}/b. \quad (3)$$

We use a fuzzy intersection operator (*t-norm*)  $\mathcal{I}$  for the “transition” from  $A_{J_1}$  to  $b$  via  $\mathcal{C}$ , including the reliability of the definition  $C_{J_3}$ , that is,

$$F_{AJ_db} = \mathcal{I}(F_{AJ_1C}, \mathcal{I}(m_{J_3C}, F_{CJ_3b})). \quad (4)$$

The same does not apply for the calculation of  $F_{AJ_da}$ , since  $A_{J_1}$  is related with  $a$  in two different ways: directly with  $F_{AJ_1a}$  and via  $C_{J_3}$  with  $\mathcal{I}(F_{AJ_1C}, \mathcal{I}(m_{J_3C}, F_{CJ_3b}))$ . We use a fuzzy union operator (*t-conorm*) to combine these two values, hence

$$F_{AJ_da} = \mathcal{U}(\mathcal{I}(F_{AJ_1C}, \mathcal{I}(m_{J_3C}, F_{CJ_3b})), F_{AJ_1a}). \quad (5)$$

Replacing  $F_{AJ_da}$  and  $F_{AJ_db}$  ((5) and (4)) in (3) produces the definition  $A_{J_d}$  of  $\mathcal{A}$ .

### 2.5. Complexity of semantic identification: an example

As an example of a semantic encyclopedia, consider using the WordNet database [15] in order to perform semantic identification in text documents. A traditional textual query is essentially a single string matching operation on the available documents. However, we may perform semantic identification on these documents if we search for strings that

imply the entity corresponding to the given string. For example, efficient retrieval of text documents related to the semantic entity “medicine” (in the sense of medical specialty) involves searching for multiple strings directly or indirectly associated with medicine, such as “medical science” (synonym), “pathology” (coordinate term), “cardiology” (hyponym), “radiotherapy” (domain term), and so forth, building a definition of the form (1). Hence, each of these semantic entities is defined by a syntactic feature (the corresponding string) and other semantic entities it is possibly associated with (such as the ones mentioned above, in the case of medicine). For this example, the number of syntactic entities that need to be evaluated in the corresponding detailed definition (i.e., the number of string matching operations that need to be performed) in order to identify “medicine” in a single text document sums up to 296 and the complexity issues involved with semantic identification become apparent. In the context of image or video identification, these string matching operations are replaced by feature extraction and image analysis algorithms, with significantly higher orders of complexity. Note that the assignment of appropriate relevance factors  $F$  to the definitions that occur is not a trivial task and if not done manually, neuro-fuzzy or soft-computing training methods may be applied to this end.

It is worth to mention that even a set of conventional algorithms used for semantic identification within image or video data can take the form of a “mini” semantic encyclopedia containing detailed definitions that involve tens of syntactic entities. An example of building a semantic tree from such a set of identification algorithms is displayed in Section 6.2.

The following sections deal with the problem of identifying semantic entities within multimedia documents and controlling the computational complexity of this process.

### 3. SEMANTIC IDENTIFICATION

#### 3.1. Modelling via automata

##### 3.1.1. Elementary automaton

Considering the simplest possible definition  $A_j$  of a semantic entity  $\mathcal{A}$  depending on a single syntactic entity  $a$ , (2) takes the form  $A_j = F_{\mathcal{A}a}/a$ . Thus, to identify the entity  $\mathcal{A}$  in a document, only employment of algorithm  $a \in \Sigma$  is required, where  $\Sigma$  denotes the entire set of algorithms under consideration and we chose to label the algorithm borrowing the name of the corresponding syntactic entity. Thus, identification of  $\mathcal{A}$  begins when the algorithm  $a$  is employed and finishes when  $a$  has run. This procedure can be represented by the automaton depicted in Figure 3(a), which we call *elementary automaton*.

##### 3.1.2. Augmented automaton

In order to represent more complex definitions (and the corresponding identification procedures), based on the evaluation of multiple syntactic properties, we use operations on elementary automata, namely intersection and union, as they

are defined in [16]. It turns out that an automaton modelling a single definition of the general form of (2) occurs after repeated intersection of all elementary automata corresponding to each single term  $F_{k,l,i}/Y_i$ . Moreover, when more than one alternative primary definitions are available, the whole process is represented by the union of the finite automata corresponding to each definition.

We call the result an *augmented automaton*, the initial state of which is followed by a number of branches that correspond to all available alternative definitions. Figure 3(b) shows the augmented automaton that corresponds to the definitions  $A_1 = \{a, b, c\}$  and  $A_2 = \{a, d\}$  of  $\mathcal{A}$  (where relevance factors are omitted for simplicity). Each state  $q$  of the automaton is labeled by an ordered pair  $(R, P)$ , where  $R$  denotes the set of algorithms run to reach  $q$  and  $P$  the algorithms pending for reaching the final state of the branch. Note that since more than one alternative definition may rely on the same syntactic properties, multiple states of the automaton may be visited simultaneously.

#### 3.2. Identification metrics

In order to design the search procedure controlling the trade-off between complexity and quality as well as to rank the results, three metrics are introduced: *complexity*, *validity*, and *certainty*.

##### 3.2.1. Complexity

The computational cost required to reach a state  $q = (R, P)$  of the augmented automaton is equal to the total complexity of the algorithms associated with the syntactic entities in  $R$ . We define as *complexity* of a state  $q = (R, P)$ , where  $R = \{t_1, t_2, \dots, t_n\}$  the syntactic entities required to reach this state, the sum of the complexities associated with these syntactic entities:

$$c((R, P)) = \sum_{t \in R} c(t), \quad (6)$$

$c(t)$  is the complexity of the algorithm  $\tau$  associated with the syntactic entity  $t$ . We also define the *total complexity* of a set of states  $Q = \{(R_1, P_1), (R_2, P_2), \dots\}$  the quantity

$$\mathcal{C}(Q) = \sum_{t \in \bigcup_j R_j} c(t) \quad (7)$$

since the same set of syntactic entities may simultaneously lead to different states in the augmented automaton.

##### 3.2.2. Validity

In every state  $q = (R, P)$  of the augmented automaton, a subset  $R$  of the syntactic entities participating in a definition has been used, hence it represents a “partial definition” of the corresponding semantic entity. The *validity* of this state measures the amount of information gathered in that state. In other words, it measures how “valid” an answer will be, if only the set  $R$  of syntactic entities is evaluated. If  $m_k$  is the

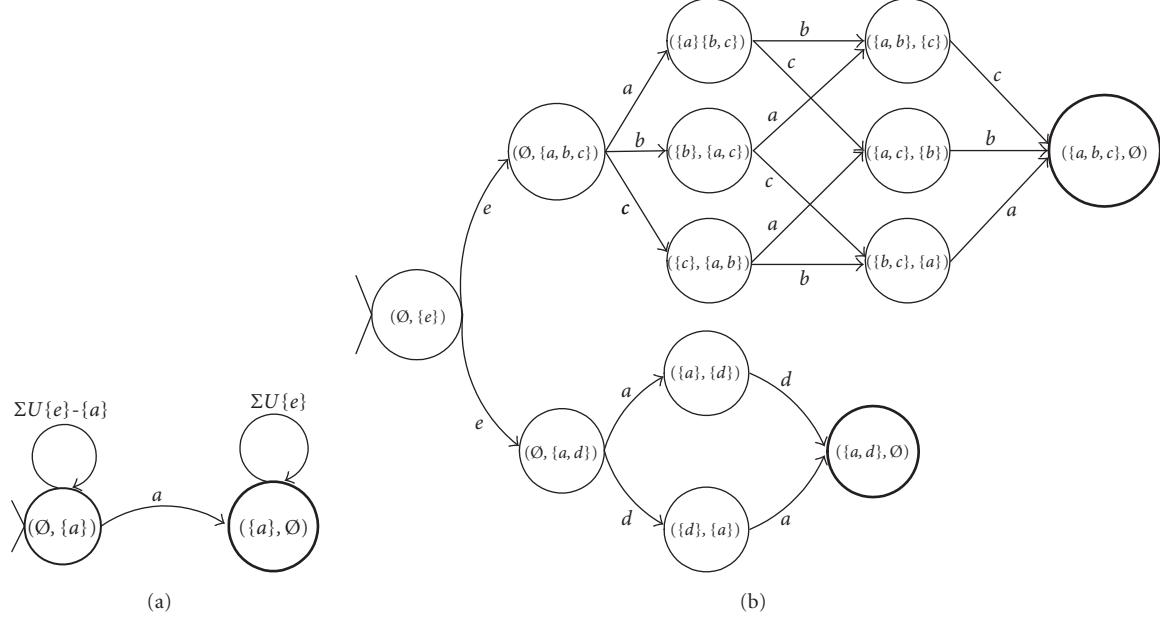


FIGURE 3: (a) The elementary automaton corresponding to definition  $A_J = F_{A_J a}/a$ . (b) The augmented automaton which corresponds to definitions  $A_1 = \{a, b, c\}$  and  $A_2 = \{a, d\}$  of  $\mathcal{A}$ . For the sake of simplicity, self-transition loops have been omitted.

reliability of the primary description  $J$  of  $E_k$ , the validity of  $q$  is defined as

$$\nu_J(E_k/q) \equiv \nu_J(E_k/(R, P)) = \mathbb{J}(m_{kJ}, \mathcal{U}_{t \in R}(F_{kjt})), \quad (8)$$

where  $t$ -conorm  $\mathcal{U}$  is deliberately used as multiple-argument operator due to its associativity property [12]. Equation (8) shows that a partial definition cannot be more valid than the complete one and that validity cannot be reduced as we traverse the automaton. Thus further consideration of syntactic entities can only increase the validity of the definition. Moreover, we define *total validity* as the fuzzy union of the validities of a set of states  $Q = \{q_1, q_2, \dots, q_n\}$ :

$$\mathcal{V}(E_k/Q) = \mathcal{U}_{q \in Q}[\nu_J(E_k/q)]. \quad (9)$$

For instance, when traversing the automaton of Figure 3(b) and having used syntactic entities  $a$  and  $b$ , we reach states  $q_1 = (\{a, b\}, \{c\})$  and  $q_2 = (\{a\}, \{d\})$ , thus  $Q = \{q_1, q_2\}$ , which yields

$$\mathcal{V}(E_k/Q) = \mathcal{U}[\mathbb{J}(m_{kJ}, \mathcal{U}(F_{kj_1 a}, F_{kj_1 b})), \mathbb{J}(m_{kJ_2}, F_{kj_2 a})]. \quad (10)$$

### 3.2.3. Certainty

Both complexity and validity depend solely on the semantic encyclopedia or, equivalently, they are independent of the data under examination. On the contrary, *certainty* is used to express the degree of our belief that a semantic entity  $E_k$  has been identified within a specific multimedia document and depends on the results of the algorithms employed to evaluate the syntactic entities appearing in the definitions of  $E_k$ .

The certainty of a state  $q = (R, P)$  is defined as

$$\mu_J(E_k/q) \equiv \mu_J(E_k/(R, P)) = \mathbb{J}(m_{kJ}, \mathcal{U}_{t \in R}(\mathbb{J}(F_{kt}, \mu_{Y_t}))), \quad (11)$$

where  $\mu_{Y_t}$  is the value of the membership function of the syntactic entity  $t$  for the specific multimedia document (see Section 2.1). Observe that for every state  $q$ , Certainty is bounded by that state's validity:

$$\mu_J(E_k/q) \leq \nu_J(E_k/q). \quad (12)$$

Similarly to the total validity of a set of states  $Q$ , its *total certainty* is

$$\mathcal{M}(E_k/Q) = \mathcal{U}_{i=1, \dots, n}[\mu_J(E_k/q_i)]. \quad (13)$$

Combining (9), (11), and (13), it turns out that

$$\mathcal{M}(E_k/Q) \leq \mathcal{V}(E_k/Q). \quad (14)$$

### 3.3. The equivalent augmented automaton

Considering an augmented automaton  $M_1$  which depicts the alternative definitions  $E_{kj_1}, E_{kj_2}, \dots, E_{kj_n}$  of an entity  $E_k$ , we then form an *equivalent augmented automaton*  $M_2$  which includes only one definition, namely,  $E_{kj} = E_{kj_1} \cup E_{kj_2} \cup \dots \cup E_{kj_n}$ . Each state  $(R, P)$  of  $M_2$  is a mapping of a set of states of  $M_1$ ,  $(R_i, P_i)$ , which we reach by running the algorithms belonging to  $A$ :

$$(R_i, P_i) = (E_{kj_i} \cap R, E_{kj_i} \cap P). \quad (15)$$

Observing that  $(R, P) = (\bigcup_i R_i, \bigcup_i P_i)$ , complexity is defined as

$$c((R, P)) = \mathcal{C}_i((R_i, P_i)). \quad (16)$$

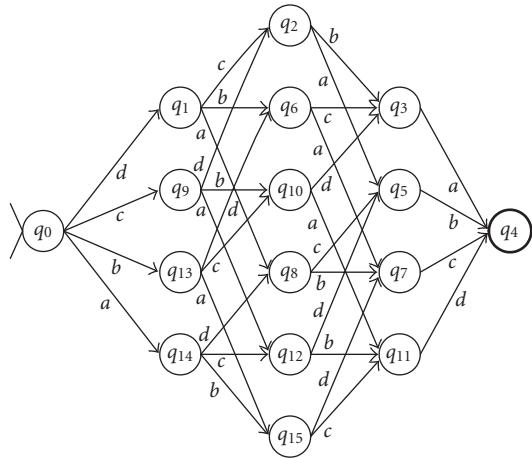


FIGURE 4: The equivalent augmented automaton representing the semantic search of  $\mathcal{A}$  as depicted in Figure 3(b). Four algorithms are employed, resulting in an automaton containing  $2^4 = 16$  states. In general, employment of  $n$  algorithms results in  $2^n$  possible states.

Moreover, taking into account (8), (9), and (15), we define as validity of  $(R, P)$  of  $M_2$  the total validity of  $(R_i, P_i)$  of  $M_1$  which, according to (9), is written as

$$v(E_k/(R, P)) = \mathcal{U}_i(v_j(E_k/(R_i, P_i))). \quad (17)$$

Finally, certainty arises in similar manner, according to (11), (13) and (15):

$$\mu(E_k/(R, P)) = \mathcal{U}_i(\mu_j(E_k/(R_i, P_i))). \quad (18)$$

Thus the equivalence between  $M_1$  and  $M_2$  stands on the fact that by running the same set of algorithms, we reach a state of the same complexity, validity, and certainty. Considering the automaton of Figure 3(b), its equivalent is depicted in Figure 4.

#### 4. DESIGN METHODOLOGIES

We next consider two possible scenarios; one where semantic search is performed under time constraints and a second where there is a strict demand for a “valid” answer.

##### 4.1. Design in terms of complexity

In the case where the identification process is constrained by a limited complexity/time budget  $C > 0$ , the search is designed as follows. First, a set  $Q_{kC}$  of all states complying with the constraint is found:

$$Q_{kC} = \{q | c(q) \leq C\}. \quad (19)$$

From the set  $Q_{kC}$ , the state  $q_0$  is chosen which provides maximum validity:

$$q_0 = \underset{q \in Q_{kC}}{\text{maximizer}} (v(E_k/q)). \quad (20)$$

As the order of execution is of no importance, it suffices to run the set of algorithms that leads to the optimal state  $q_0$ .

##### 4.2. Design in terms of validity

Considering the inverse problem where we seek for “valid” identification, there applies a validity threshold  $V$ , under which no answer is accepted. In this case, the corresponding set  $Q_{kV}$  satisfying this criterion is selected:

$$Q_{kV} = \{q | v(E_k/q) \geq V\}. \quad (21)$$

Next, the state which requires the less complexity is chosen:

$$q_0 = \underset{q \in Q_{kV}}{\text{minimizer}} (c(q)). \quad (22)$$

Once again, the order of execution of the algorithms plays no role.

#### 5. OPTIMIZATION USING DYNAMIC PROGRAMMING

When modelling the identification procedure of a semantic entity  $E_k$  with a single detailed definition  $E_{kj_d}$  of the form of (2) as  $E_{kj_d} = F_{kj_d1}/Y_1 + \dots + F_{kj_dn}/Y_n$ , the resulting automaton has  $2^n$  states, that is, its size grows exponentially with respect to the number of involved syntactic properties. Consequently, minimization required in (20) or (22) performed by exhaustive evaluation of  $c(q)$  for all  $q$  is practically impossible.

In order to overcome this issue, we convert this optimization task into a “knapsack problem” [17]. We notice that when passing over from one state to another by invoking an algorithm (i.e., evaluating a syntactic property), both complexity and validity grow in a nondecreasing manner. The problem of Section 4.1 can be redefined in the following form.

Given a collection  $\{1, 2, \dots, n\}$  of  $n$  algorithms and a complexity threshold  $C > 0$ , the goal is to find the optimal set  $X^*(C) = \{x_1, x_2, \dots, x_n\}$ , with  $x_i \in \{0, 1\}$  denoting whether algorithm  $i$  has been invoked ( $x_i = 1$ ) or not ( $x_i = 0$ ), so as to maximize the quantity

$$\mathcal{U}_{i=1,\dots,n}(F_{kj_i} \cdot x_i) \quad (23)$$

under the restriction

$$\sum_{i=1}^n (c(t_i) \cdot x_i) \leq C. \quad (24)$$

This is a variant of the knapsack problem, with a nonlinear profit function (validity) which can be efficiently solved by the use of dynamic programming. Such an approach solves not only the original problem  $X_n(C)$  of computing maximum validity  $\text{val}_n(C)$  under the threshold  $C$ , using  $n$  algorithms, but also all subproblems of the form  $X_j(d)$ , where  $d = 0, \dots, C$  and  $j = \{1, \{1, 2\}, \dots, \{1, 2, \dots, n\}\}$  (with algorithms ordered in a random way). In fact, this process returns an array containing the solutions  $\text{val}^*(d)$ ,  $d = 0, \dots, C$  for all subproblems and  $\text{val}^*(d) \equiv \text{val}_n(d)$ , where all  $n$  algorithms are taken into account. The respective states are  $X^*(d) \equiv X_n(d)$ .

We use the Bellman recursion [18] where if  $\text{val}_{j-1}(d)$  has been computed for all complexity thresholds  $d = 0, \dots, C$ ,

```

(1)   for  $d := 0$  to  $C$ 
(2)      $\text{val}_0(d) := 0$ 
(3)   for  $j := 1$  to  $n$ 
(4)     for  $d := 0$  to  $c(j) - 1$ 
(5)        $\text{val}_j(d) := \text{val}_{j-1}(d)$ 
(6)       for  $d := c(j)$  to  $C$ 
(7)         if  $\mathcal{U}(\text{val}_j(d - c(j)), F_{kj}) > \text{val}_{j-1}(d)$ 
(8)            $\text{val}_j(d) := \mathcal{U}(\text{val}_j(d - c(j)), F_{kj})$ 
(9)         else
(10)         $\text{val}_j(d) := \text{val}_{j-1}(d)$ 
(11)   for  $d := 0$  to  $C$ 
(12)      $\text{val}^*(d) := \text{val}_n(d)$ 

```

ALGORITHM 1: Designing with dynamic programming.

then we can calculate  $\text{val}_j(d)$  (i.e., consider algorithm  $j$ ) as

$$\text{val}_j(d) = \begin{cases} \text{val}_{j-1}(d) & \text{if } d \leq c(j), \\ \max \{\text{val}_{j-1}(d), \mathcal{U}(\text{val}_j(d - c(j)), F_{kj})\} & \text{if } d \geq c(j). \end{cases} \quad (25)$$

The pseudocode of this process is presented in Algorithm 1.

In order to compute the optimal sets of algorithms (states)  $X^*(d)$  for each threshold, we observe (see [17, pages 24–25]) that the set  $X_j(d)$  in iteration  $j$  differs from the previous set  $X_{j-1}(d)$  by at most one item ( $x_j$ ). It is thus sufficient to keep, in every iteration, a pointer  $A_j(d) \in \{0, 1\}$ , meaning that

$$A_j(d) = \begin{cases} 1 & \text{if } \text{val}_j(d) := \mathcal{U}(\text{val}_j(d - c(j - 1)), F_{kj}), \\ & \quad \text{algorithm } j \text{ was included}, \\ 0 & \text{if } \text{val}_j(d) := \text{val}_{j-1}(d), \\ & \quad \text{algorithm } j \text{ was not included}. \end{cases} \quad (26)$$

Finally, to reconstruct the set  $X^*(d)$  by going through the pointers, we do the following. If  $A_n(d) = 1$ , then algorithm  $n$  was included in the optimal set and we go on by checking  $A_{n-1}(d - c(j))$ , otherwise ( $A_n(d) = 0$ )  $n$  was not included and we proceed with  $A_{n-1}(d)$ .

This algorithm computes  $X^*(d)$  and  $\text{val}^*(d)$  in pseudopolynomial time, with worst-case complexity of  $O(nC)$ , significantly reducing the computational cost, compared to the original  $O(2^n)$ . Since all subproblems are solved, it suffices to run this algorithm once, by using  $C_{\text{tot}} = \sum_{i=1}^n (c(t_i))$  as threshold and compute the optimal states for various thresholds within the space  $[0, \dots, C_{\text{tot}}]$ . Thus, if we choose a dense partitioning of this space, we can practically design every search in terms of complexity in constant time (by using the resulting array).

Inversely, when designing in terms of validity, a solution can be found by selecting the first state of the precomputed array that satisfies the criterion. This process must be run for all branches of the augmented automaton. For more information on solving knapsack problems, one should refer to [19].

## 6. EXAMPLES AND EXPERIMENTAL EVALUATION

In the following, a set of experiments and examples is presented to display the effectiveness of the proposed method. In the first experiment, a simplified definition of the entity “table” is employed and is identified within a set of sketches. The next experiment tries to identify track and field running within a video sequence consisting of three types of sports and finally the last experiment uses random input (random weights and complexities in definitions) following different distributions and calculates the gain in complexity versus validity for various threshold values and design methods.

### 6.1. Identifying a “table”

For the first experiment, we create a definition for the entity table as shown in Figure 5. To produce the detailed definition, we use the algebraic product  $\mathcal{I}(a, b) = ab$  as a fuzzy intersection operator and its dual, the algebraic sum  $\mathcal{U}(a, b) = a + b - ab$ , for union. Composing the two definitions as described in Section 2.4, the following primary definition of “table” is obtained:

$$E_{01} = 0.9/Y_{01} + 0.378/Y_{02} + 0.567/Y_{03} + 0.567/Y_{04}. \quad (27)$$

For each syntactic property, an estimate of its complexity was experimentally obtained as shown in Table 1. Complexity units in this list correspond to  $10^3$  flops. We used the drawings illustrated in Figure 6 to perform the semantic search and calculate the degree (certainty) up to which each drawing contains the entity “table,” as defined above. Results of design in terms of validity are illustrated in Table 1. Two comments are worth to be made. (1) Modifying  $V$  results in selection of different algorithms (see, e.g., rows one and two). (2) Relatively high validity and certainty is obtained at reasonably low computational cost, but pushing the validity threshold to its high levels causes abrupt increase of the required complexity.

Similarly, results of design in terms of complexity have been included in Table 2 for complexity bounds  $C = 3, 7, 8, 13, 7$ . Commenting on these results, decent validity levels are attained even under low complexity constraints. Allowing higher complexity budgets enhances both validity and certainty but the gained increase is not proportional to the additional computational cost.

The results from both design strategies indicate that efficient policies can be adopted for optimal use of resources by balancing between complexity and validity.

### 6.2. Sports identification example

A widely adopted paradigm used for demonstrating semantic search techniques is sports identification and annotation. A variety of such techniques has been presented in the bibliography (see, e.g., [20–22]). Present work does not intent to fully confront this problem, but to display how our methodology can be utilized for optimizing existing procedures in terms of computational cost.

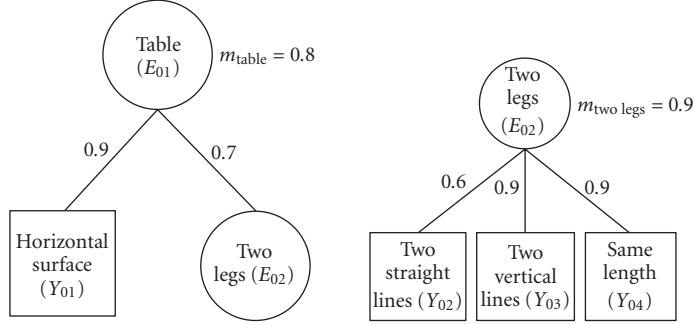


FIGURE 5: Description of the entity “table.”

TABLE 1: Design in terms of validity. Changing the threshold results in selection of different sets of algorithms.

V	Val.	Compl.	Algorithms	(a)	(b)	(c)	(d)	(e)	(f)
0.2	0.45	3.3	4	0.4	0.09	0	0.438	0.432	0.403
0.46	0.72	3.6	1	0.705	0.70	0.675	0.396	0.188	0.685
0.73	0.77	6.9	1, 4	0.753	0.703	0.675	0.617	0.519	0.743
0.785	0.79	16.2	1, 2, 3, 4	0.782	0.763	0.675	0.73	0.64	0.765

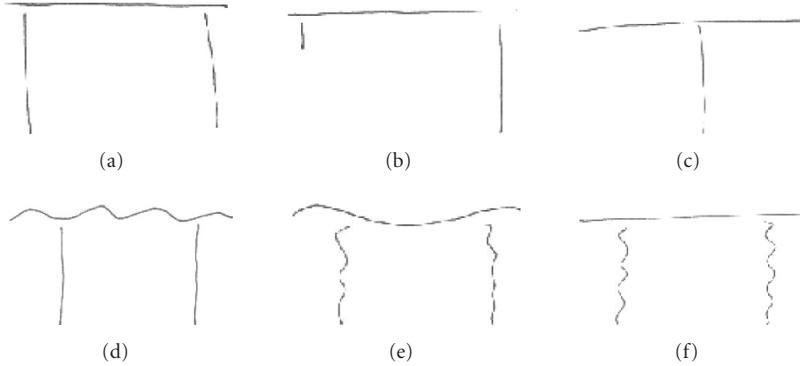


FIGURE 6: Drawings of a “table.”

TABLE 2: Design in terms of complexity.

C	Val.	Compl.	Algorithms	(a)	(b)	(c)	(d)	(e)	(f)
3.7	0.72	3.6	1	0.705	0.7	0.675	0.396	0.188	0.685
8	0.7654	6.9	1, 4	0.753	0.703	0.675	0.617	0.519	0.743
13	0.7827	11.4	1, 3, 4	0.771	0.746	0.675	0.692	0.572	0.752
17	0.79	16.2	1, 2, 3, 4	0.782	0.763	0.675	0.73	0.64	0.765

In this experiment, we use a 1181-frame video sequence consisting of three types of sports, namely soccer, track and field running, and swimming (Figure 7 shows three frames of the sequence). Table 3 shows which frames correspond to which sport. By using the definitions

$$\begin{aligned} E_{00J_1} &= 0.9/E_{01} + 0.5/Y_{00}, \\ E_{01J_2} &= 0.9/Y_{01} + 0.7/Y_{02}, \end{aligned} \quad (28)$$

and ignoring their reliability (by setting  $m = 1$ ) for simplicity, we identify the frames which display running. The entities represented by the above symbols are shown in Table 4, along with their complexities, where applicable.

The values of the relevance factors (weights) denote the importance of each syntactic entity in the identification. For example, note that the syntactic entity “Bordeaux ground color” ( $Y_{01}$ ) has a value of  $F_{02J_{101}} = 0.9$ , that is, a high relevance factor. This choice is justified by the importance of



FIGURE 7: Frames from the three video sequences used.

TABLE 3: Sports and the corresponding frames.

Sport	Frame range
Soccer	1–512
Running	513–811
Swimming	812–1181

TABLE 4: Symbols, entities, and the corresponding complexities.

Symbol	Entity	Complexity (kflops)
$E_{00}$	Track and field running	N/A
$E_{01}$	Track and field running terrain	N/A
$Y_{00}$	Parallel mobile object movement	20
$Y_{01}$	Bordeaux ground color	8
$Y_{02}$	8 straight lines	15



FIGURE 8: The straight-lines algorithm result. Note that some of the lines occur from the environment and not the terrain.

color in distinguishing the terrain, as well as the robustness of the algorithm involved. On the other hand, the syntactic entity “8 straight lines” has a lower weight associated with it, even though this entity is very important when identifying the terrain as well. The reason for this is that the method used for line extraction (Hough transform) usually identifies extra straight lines which exist in the environment, apart from the lines of the terrain (Figure 8), yielding less accurate results. As for the syntactic entity  $Y_{03}$  “parallel mobile object movement,” the method presented in [23] was employed.

In order to obtain the detailed definition of the entity “track and field running,” we use the same fuzzy operators

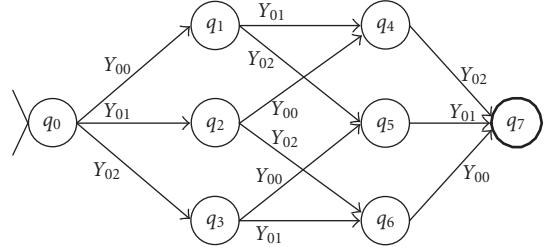


FIGURE 9: Automaton representing the identification of track and field running.

as in the previous experiment (Section 6.1), resulting in

$$E_{00J_d} = 0.5/Y_{00} + 0.81/Y_{01} + 0.63/Y_{02}. \quad (29)$$

Execution of all the algorithms (without any optimization method applied) successfully identifies the part of the video sequence displaying track and field running. Still, it is possible to reduce the required complexity of the identification while maintaining satisfactory results, if we choose to use only a subset of the three syntactic entities. Table 5 shows the complexity, validity, and mean certainty value (for those frames displaying running and those that do not) obtained by evaluating all possible combinations of syntactic entities, or equivalently, visiting all states of the corresponding automaton, which is shown in Figure 9.

Figure 10 demonstrates the attained certainty values for the identification of track and field running  $E_{00}$  in each frame of the video sequence. The three diagrams show the results under different complexity thresholds that have led to states  $q_2$ ,  $q_6$ , and  $q_7$  with respective complexities 8, 23, and 43. Commenting on these, one can see that the increase of complexity budget leads to better identification results. However, decent identification of the entity is achieved even when having employed fewer algorithms (states  $q_2$  and  $q_6$ ).

### 6.3. Experiments with random input

A fully synthetic experiment was set up in order to evaluate our method given a variety of randomly distributed relevance factors and algorithm complexities. We present here the design results using two detailed definitions consisting of 30 syntactic properties each. Figures 11(a) and 11(b) illustrate the design results in terms of complexity

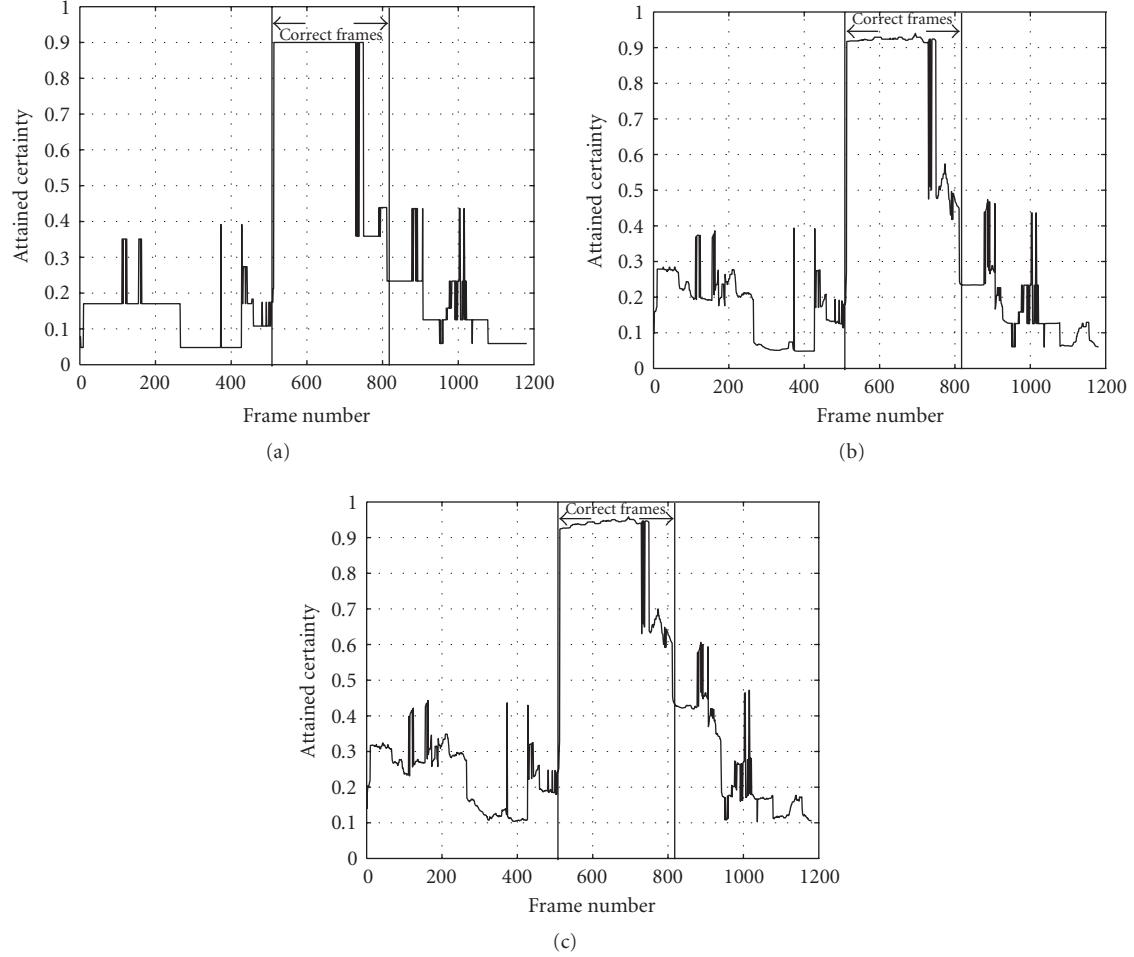
FIGURE 10: Identification results for (a) state  $q_1$ , (b) state  $q_6$ , and (c) state  $q_7$ .

TABLE 5: Results for all states of the automaton.

State	Evaluated properties	Complexity (kfllops)	Validity	Mean certainty (running frames)	Mean certainty (other frames)
$q_0$	—	0	0	0	0
$q_1$	$Y_{00}$	20	0.5	0.2532	0.1609
$q_2$	$Y_{01}$	8	0.81	0.7838	0.2429
$q_3$	$Y_{02}$	15	0.63	0.2205	0.0542
$q_4$	$Y_{00}, Y_{01}$	28	0.905	0.8436	0.377
$q_5$	$Y_{00}, Y_{02}$	35	0.815	0.4174	0.2103
$q_6$	$Y_{01}, Y_{02}$	23	0.9297	0.8269	0.2887
$q_7$	$Y_{00}, Y_{01}, Y_{02}$	43	0.96485	0.8749	0.4188

and validity, respectively, for various thresholds. Complexities and relevance factors of the first definition follow uniform distributions in [1, 10] and [0, 0.5], respectively, while for the second definition, normally distributed values were used. The mean values were 5 and 0.3, while the standard deviation was 0.1 and 0.01 for complexities and relevance factors.

These results confirm the remarks made on the first experiment (Section 6.1) regarding the gain of complexity

versus validity using the proposed method. Furthermore, from the experiments conducted, we observed that the method is very efficient when dealing with widely distributed values, contrary to the case of distributions with low variance. One could expect this, since algorithms with similar values of complexities (or weights of the corresponding syntactic entities) make the selection between them unimportant (i.e., they can be considered “equivalent” under our scope).

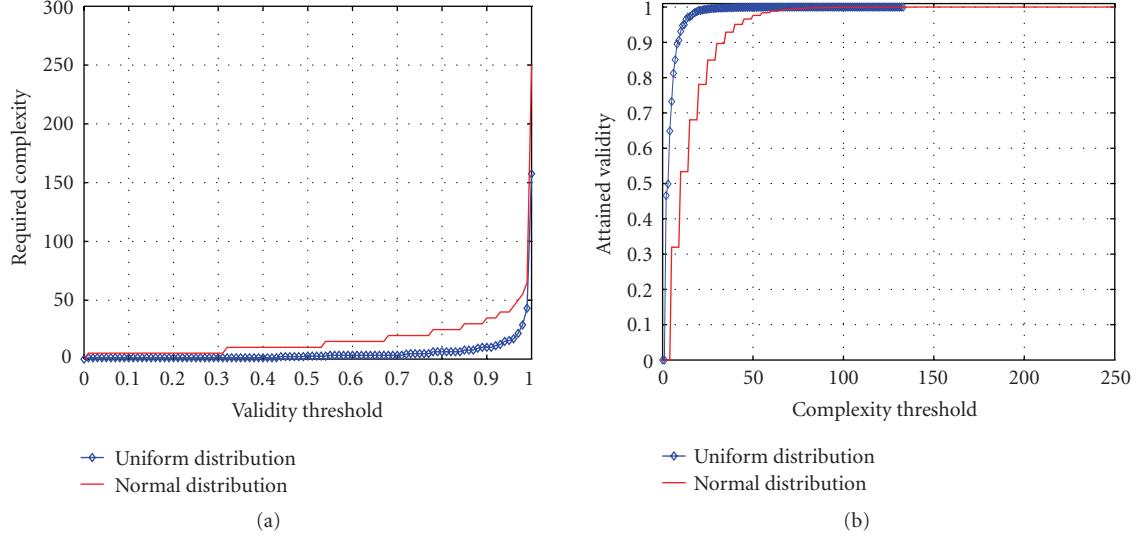


FIGURE 11: (a) Design in terms of validity. It is interesting to observe that validity of about 0.8 is attained for only about 10% of the total complexity. (b) Design in terms of complexity.

## 7. CONCLUSIONS AND FUTURE WORK

In this work, a methodology was presented which can act as a mechanism for controlling the computational cost (complexity) of semantic identification while retaining the quality of the result (validity). Our approach could be particularly useful in time-critical applications (real-time or bulky processing). The methodology relies on the mathematical model of a semantic encyclopedia which can be used for identification of semantic entities in multimedia documents. The encyclopedia utilizes fuzzy sets to efficiently model the uncertainty that is associated with high-level vague concepts and is built on a hierarchical structure. Finite automata were used to model the identification design, with each automaton state being triggered by the execution of an algorithm. Dynamic programming was used to reduce the originally exponential number of states that need to be taken into account in order to find the optimal solution. Real life as well as synthetic experiments prove that this method can optimize the search under complexity and validity constraints.

This paper does not address the problem of constructing the semantic encyclopedia. However, automatic annotation techniques already presented in the bibliography can be remodelled using the semantic encyclopedia approach, thus enabling complexity control of the annotation and search procedures.

Some remarks are worth to be made concerning the structure of the semantic encyclopedia, as well as the use of the algorithms and syntactic entities. Identification results rely on the amount of information contained in the respective definitions. In the modelling process, it is important that the algorithms employed by these methods are properly evaluated and assigned suitable weights to the corresponding definitions. In this case, the validity that occurs in a state of the search allows us to “trust” the certainty of

the identification. This means that we expect that a high certainty value occurs wherever the entity actually exists.

Future work includes the extension of the capabilities of the semantic encyclopedia by use of mathematical logics and a variety of relations (e.g., position, inclusion, etc.,) that would significantly enhance its expressive power.

## REFERENCES

- [1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, “Matching words and pictures,” *Journal of Machine Learning Research*, vol. 3, no. 7, pp. 1107–1135, 2003.
- [2] M. Wallace, Y. Avrithis, G. Stamou, and S. Kollias, “Knowledge-based multimedia content indexing and retrieval,” in *Multimedia Content and Semantic Web: Methods, Standards and Tools*, G. Stamou and S. Kollias, Eds., John Wiley & Sons, New York, NY, USA, May 2005.
- [3] A. Dorado and E. Izquierdo, “Semantic labeling of images combining color, texture and keywords,” in *Proceeding of IEEE International Conference on Image Processing (ICIP ’03)*, vol. 3, pp. 9–12, Barcelona, Spain, September 2003.
- [4] M. S. Lew, “Next-generation web searches for visual content,” *IEEE Computer*, vol. 33, no. 11, pp. 46–53, 2000.
- [5] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, New York, NY, USA, 2002.
- [6] T. Sikora, “The MPEG-7 visual standard for content description—an overview,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 696–702, 2001.
- [7] R. Visser, N. Sebe, and M. S. Lew, “Detecting automobiles and people for semantic video retrieval,” in *Proceeding of 16th International Conference on Pattern Recognition (ICPR ’02)*, vol. 2, pp. 733–736, Quebec City, Canada, August 2002.
- [8] P. Duygulu, K. Barnard, N. de Freitas, and D. A. Forsyth, “Object recognition as machine translation: learning a lexicon for a fixed image vocabulary,” in *Proceeding of 7th European Conference on Computer Vision (ECCV ’02)*, vol. 4, pp. 97–112, Copenhagen, Denmark, May 2002.

- [9] G. Akrivas, G. B. Stamou, and S. Kollias, "Semantic association of multimedia document descriptions through fuzzy relational algebra and fuzzy reasoning," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 34, no. 2, pp. 190–196, 2004.
- [10] M. Wallace and S. Kollias, "Computationally efficient incremental transitive closure of sparse fuzzy binary relations," in *Proceeding of IEEE International Conference on Fuzzy Systems (IEEE-FUZZ '04)*, Budapest, Hungary, July 2004.
- [11] Y. Avrithis, G. Stamou, M. Wallace, et al., "Unified access to heterogeneous audiovisual archives," *Journal of Universal Computer Science*, vol. 9, no. 6, pp. 510–519, 2003.
- [12] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall, Upper Saddle River, NJ, USA, 1995.
- [13] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, New York, NY, USA, 2003.
- [14] U. Straccia, "Reasoning within fuzzy description logics," *Journal of Artificial Intelligence Research*, vol. 14, pp. 137–166, January–June 2001.
- [15] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Mass, USA, 1998.
- [16] H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, Upper Saddle River, NJ, USA, 1998.
- [17] H. Kelleler, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer, Berlin, Germany, 2004.
- [18] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.
- [19] K. M. Brethauer and B. Shetty, "The nonlinear knapsack problem—algorithms and applications," *European Journal of Operational Research*, vol. 138, no. 3, pp. 459–472, 2002.
- [20] J. Assfalg, M. Bertini, C. Colombo, and A. Del Bimbo, "Semantic annotation of sports videos," *IEEE Multimedia*, vol. 9, no. 2, pp. 52–60, 2002.
- [21] R. Leonardi, P. Migliorati, and M. Prandini, "Semantic indexing of sports program sequences by audio-visual analysis," in *Proceeding of IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 9–12, Barcelona, Spain, September 2003.
- [22] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden Markov models," *Pattern Recognition Letters*, vol. 25, no. 7, pp. 767–775, 2004.
- [23] G. Tsechpenakis, Y. Xirouchakis, and A. Delopoulos, "Main mobile object detection and localization in video sequences," in *Proceeding of 4th International Conference on Advances in Visual Information Systems (VISUAL '00)*, vol. 1929 of *Lecture Notes in Computer Science*, pp. 84–95, Lyon, France, November 2000.

**M. Falelakis** received his Diploma from the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003. He is currently a Ph.D. candidate with the same department. His research interests include semantic identification techniques, content-based retrieval of multimedia, knowledge representation, and fuzzy inference. He is a Member of the Technical Chamber of Greece.



**C. Diou** received his Diploma in electrical and computer engineering from Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2004. He is currently pursuing his Ph.D. degree at the same university. His research interests include content-based retrieval of multimedia data, semantic knowledge extraction and representation, as well as image and video analysis. He is a Member of the Technical Chamber of Greece.



**A. Delopoulos** was born in Athens, Greece, in 1964. He graduated from the Department of Electrical Engineering of the National Technical University of Athens (NTUA) in 1987, received the M.S. degree from the University of Virginia in 1990 and the Ph.D. degree from NTUA in 1993. He is with the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki, where he serves as an Assistant Professor. His research interests lie in the areas of multimedia data understanding and computer vision. He is the author of more than 60 journals and conference scientific papers. He has participated in 17 European and national R&D projects related to application of signal, image, video, and information processing to entertainment, culture, education, and health sectors. He is a Member of the Technical Chamber of Greece and IEEE.

