Complexity Control in Semantic Identification

M. Falelakis, C. Diou, A. Valsamidis and A. Delopoulos
Multimedia Understanding Group, Information Processing Laboratory
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki, Greece
Email: {manf, diou, tasos}@olympus.ee.auth.gr, adelo@eng.auth.gr

***Abstract*— This paper proposes a methodology for modeling the process of semantic identification and controlling its complexity and accuracy of the results. Each semantic entity is defined in terms of lower level semantic entities and low level features that can be automatically extracted, while different membership degrees are assigned to each one of the entities participating in a definition, depending on their importance for the identification. By selecting only a subset of the features that are used to define a semantic entity both complexity and accuracy of the results are reduced. It is possible, however, to design the identification using the metrics introduced, so that satisfactory results are obtained, while complexity remains below some required limit.**

I. Introduction

The vast amount of data in multimedia collections and the need for better human-computer interaction in modern applications require the content management of multimedia data at a *semantic* level. Recent standardization efforts such as the MPEG-7 [1] provide the means to describe content while algorithms and techniques are introduced that aim at the automatic extraction of semantic information from multimedia documents (see, for example [2] and [3]). This paper proposes a general model which can be adopted by such techniques in order to achieve higher expressive capabilities in the semantic analysis process, but mainly to achieve complexity control so that the analysis can cope with the requirements and limitations of real life applications.

Identification of a semantic entity (an event, an object etc) is essentially equivalent to the computation of the *degree* up to which the entity exists in a multimedia document. Lower level features are used to assess this degree and based on the assumption that often some features are more important than others, we use metrics that allow us to estimate the validity of the identification process if only a subset of the features is used. Hence we may obtain satisfactory results while at the same time the computational cost is reduced.

The Fuzzy Semantic Encyclopedia acts as a knowledge base providing definitions of high level entities using lower level features and is presented in section II. The metrics required to measure the existence of an entity in a document i.e., the *certainty* of the identification, the *validity* of this process as well as its *complexity* are presented in section III. Design methods that optimize the identification with limited resources are also presented in the same section. While section III refers to semantic entities related directly to syntactic properties, section IV extends these results to the identification of entities containing other (lower level) semantic entities in their definitions. Three types of substitution procedures are proposed to cope with this scenario. Examples and experiments displaying and evaluating the proposed method are presented in section V while section VI includes remarks and future perspectives of this work.

II. Fuzzy Semantic Encyclopedia

As mentioned above, semantic analysis relies on the definition of higher level entities in terms of lower level syntactic features of the document. Such features are quantities that machines are able to compute and can be viewed as the alphabet that is used to form definitions of high level entities in the same sense that letters form words. These definitions construct a knowledge base that we call *Fuzzy Semantic Encyclopedia* and is presented in this section. The structural elements of the Encyclopedia are the *Syntactic* and *Semantic Entities*.

A. Syntactic Entities

As *syntactic feature* t we define any measurable quantity (e.g., brightness, frequency) that can be obtained by applying a corresponding algorithm τ on the given data set (e.g., on a scene, an image or a signal). For simplicity, we assume syntactic features with real values, 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 on a syntactic feature t. For example the property "very bright" is defined on the feature "brightness" and the property "red" is defined on the feature "color". We assign a 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. In general, if t_τ is the outcome of algorithm τ that "measures" t, the membership value $\mu_{Y_i} \equiv y_i(t_\tau)$ corresponds to the *degree* that the particular data set assumes property Y_i.

B. Semantic Entities

As the name implies, the term *Semantic Entity* refers to higher level objects or concepts that cannot be directly

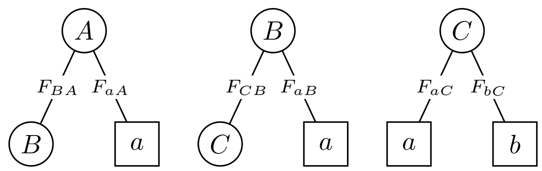

Fig. 1. Definitions of three Semantic Entities in a graph representation. Note that B is indirectly related with the Syntactic Entities in C and similarly A is related with the Syntactic Entities in both B and C.

measured and are closer to human perception. Each Semantic Entity is assigned a label $E_k \in \mathbf{E}$, where $\mathbf{E}$ is the set of all Semantic Entities considered. The Semantic Encyclopedia is built on the assumption that a Semantic Entity E_k can be described using other lower level Semantic as well as Syntactic Entities which collectively form the *scope* $\mathbf{S}_{E_k}$ of E_k. Any scope $\mathbf{S}_{E_k}$ is a subset of $\mathbf{S} = \mathbf{Y} \cup \mathbf{E}$, the set of all Semantic and Syntactic Entities. For example, one may describe Entity A with scope $\mathbf{S}_A = \{a, B\}$, where B is in turn described using $\mathbf{S}_B = \{C, a\}$ and C with $\mathbf{S}_C = \{a, b\}$ as Figure 1 illustrates. Notice that lowercase and uppercase letters denote Syntactic and Semantic Entities respectively.

Descriptions of this form simply state that existence of a Syntactic or Semantic Entity implies the existence of a higher level Semantic Entity (a and B imply the existence of A, in the example above). This is equivalent to assuming inference propositions of the form

$$p_{S_i E_k} : S_i \Rightarrow E_k, \text{ for all } S_i \in \mathbf{S}_{E_k}.$$

However in most cases inference is valid up to a certain degree $F_{S_i E_k} \in [0, 1]$ quantifying our belief regarding the truth of each proposition $p_{S_i E_k}$. In that sense each description corresponds to a discrete fuzzy set of the form

$$E_k = F_{1k}/S_1 + F_{2k}/S_2 + \ldots + F_{nk}/S_n, \tag{1}$$

where $F_{ik} \equiv F_{S_i E_k}$, that is called a *definition* of E_k. In the definition trees of Figure 1 the degrees $F_{S_i E_k}$ are presented as weights to the edges of each tree.

Definitions of the form of Equation (1) that are included in the Encyclopedia are called *primary definitions* and as explained, may contain both Syntactic and Semantic Entities. For a definition that is based only on Syntactic Entities we reserve the term *detailed definition* and is of the form

$$E_k = F_{1k}/Y_1 + F_{2k}/Y_2 + \ldots + F_{nk}/Y_n, \tag{2}$$

where $Y_i \in \mathbf{Y}$. Any non-detailed primary definition can be transformed into a detailed one, using a substitution procedure, as shown in section IV-A.

Some comments on the Encyclopedia are worth to be made. The expertise of the system (what we assume that an "expert" provides) is essentially the values of the weights F. As for the expressive capabilities of the Encyclopedia in defining Semantic Entities, we note that these strongly depend on the number of features that are used to extract low level information and the accuracy of the corresponding algorithms.

III. Semantic Inference

On the basis of the definitions included in the Semantic Encyclopedia, it is possible to perform the actual search by evaluating the values μ_{Y_i} of the Syntactic Entities involved. This is achieved by using metrics that quantify the existence of Semantic Entities, given the algorithm results.

A. Metrics of Identification

Evaluation of a Syntactic Entity Y_i participating in a detailed definition is equivalent to running its corresponding algorithm τ and computing the membership degree μ_{Y_i} up to which the document under examination assumes property Y_i, as stated in section II-A. In a similar manner, a metric is defined that denotes the degree up to which a Semantic Entity exists in a document and is called *Certainty* of the identification. Given a detailed definition of a Semantic Entity E_k in the form of Equation (2) and the membership degrees μ_{Y_i} of the Syntactic Entities Y_i in a specific document, Certainty that E_k exists in that document is defined as

$$\mu_{E_k} \stackrel{\triangle}{=} \mathop{\mathcal{U}}_i (\mathcal{I}(F_{Y_i E_k}, \mu_{Y_i})) \tag{3}$$

where the operators $\mathcal{U}$ and $\mathcal{I}$ denote fuzzy union and intersection operators respectively.

The maximum possible value of μ_{E_k} is assigned the term *Validity* of the definition and is equal to

$$\mathcal{V}(E_k) \stackrel{\triangle}{=} \mathop{\mathcal{U}}_i (F_{Y_i E_k}), \tag{4}$$

attained for $\mu_{Y_i} = 1$ for all Y_i in the scope of E_k and the use of the identity $I(a, 1) = a$ (true for every t-norm $\mathcal{I}$).

Validity denotes the maximum amount of information that a definition can provide and is used extensively in the identification design process, explained in III-D. We must note that Validity is independent of the data set under examination and can be computed *prior* to the identification. Validity is therefore a property of the definition itself.

Another characteristic of a definition is the computational complexity associated with the algorithms corresponding to its Syntactic Entities. We assign a computational cost $c(t)$ to every syntactic feature t that is essentially equal to the cost of it's corresponding algorithm τ. Hence, we may now define *Complexity* of a definition as

$$\mathcal{C}(E_k) = \sum_i c(t_i) \tag{5}$$

where t_i are the syntactic features required to evaluate the properties Y_i of the definition E_k. Notice that this value will normally depend on the size of the input data, as will the values $c(t_i)$. At least, though, worst or average case expressions of $c(t_i)$ can be considered as independent of the actual content of the examined data sets. In this perspective $\mathcal{C}(E_k)$ is also computable prior to identification.

Certainty, Complexity and Validity have been introduced for detailed definitions so far, but these metrics are applied to any definition, since each one of them can be transformed into a detailed one. Moreover, once the Certainty of an Entity is available, the latter can be treated in the same way a Syntactic Entity is treated in a detailed definition. All the previous are shown in detail in section IV, but for mathematical convenience we first deal with the expression of definitions using fuzzy relations, as well as with the design of the identification process for detailed definitions.

B. Representation of Definitions using Fuzzy Relations

A fuzzy relation that provides the values of F for the elements in the scope of a Semantic Entity E_k provides all required information regarding E_k. In this case, each value F can be considered as an element of a fuzzy relation on $\mathbf{S} \times \mathbf{S}$. We are therefore able to represent definitions using fuzzy relations and operations among them using operations between fuzzy relations. As an example, the definition A of Figure 1 corresponds to

$$\mathbf{R}_A = \begin{array}{c c} & \begin{array}{ccccc} A & B & C & a & b \end{array} \\ \begin{array}{c} A \\ B \\ C \\ a \\ b \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ F_{BA} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ F_{aA} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{6}$$

Note that definitions correspond to reflexive relations since each Entity fully implies itself. If in addition we collect the degrees μ_{Y_i} of all Syntactic Entities within the scope of E_k into a fuzzy set $\mathbf{X}$ of the form $\mathbf{X} = [0 \ldots 0 \, \mu_{Y_1} \ldots \mu_{Y_n}]$ we may compute the Certainty μ_{E_k} using

$$\mathbf{Z} = [0 \ldots \, \mu_{E_k} \ldots 0 \, \mu_{Y_1} \ldots \mu_{Y_n}] = \mathbf{X} \circ \mathbf{R}_{E_k} \tag{7}$$

In this case, the composition "$\circ$" is a generalization of the well known $\sup$-t composition (see [4] for example) and is defined as $(\mathbf{A} \circ \mathbf{B})(i,j) = \mathop{\mathcal{U}}_{k}(\mathcal{I}(a_{ik}, b_{kj}))$ where the operators $\mathcal{U}$ and $\mathcal{I}$ can be any t-conorm and t-norm respectively. By using "$\circ$" Certainty μ_{E_k} provided by (7) is equal to that of Equation (3). Validity can be computed using (7) as well, by setting $\mu_{Y_i} = 1$ for every $Y_i \in \mathbf{Y}$.

C. Partial Evaluation

Having limited resources, i.e. a limited complexity "budget" for the identification of a Semantic Entity in a data set, one must be able to choose a subset of the algorithms that are used to define the Entity. These algorithms should provide the best results for the identification, while keeping the complexity below the limit. To this purpose, partial Certainty, Validity and Complexity are defined in this section.

Assume that a subset $\mathbf{A} \subseteq \mathbf{A}_t$ of the set of Syntactic Entities $\mathbf{A}_t = \mathbf{S}_{E_k} \cap \mathbf{Y}$ participating in the detailed definition E_k are used for the identification. Then, *Partial Certainty* of the identification is defined as

$$\mu_{E_k}(\mathbf{A}) \triangleq \mathop{\mathcal{U}}_{Y_i \in \mathbf{A}} (\mathcal{I}(F_{Y_i E_k}, \mu_{Y_i})) \tag{8}$$

and denotes the confidence we have acquired that E_k exists in a data set by evaluating only the properties in $\mathbf{A}$. By Equations (3) and (8) and the monotonicity of the fuzzy union operators, we conclude that if $\mathbf{A} \subseteq \mathbf{A}'$, then $\mu_{E_k}(\mathbf{A}) \leq \mu_{E_k}(\mathbf{A}')$ and particularly $\mu_{E_k}(\mathbf{A}) \leq \mu_{E_k}(\mathbf{A}_t) \equiv \mu_{E_k}$.

Similarly, the *Partial Validity* is defined as

$$\mathcal{V}(E_k/\, \mathbf{A}) \triangleq \mathop{\mathcal{U}}_{Y_i \in \mathbf{A}} (F_{Y_i E_k}). \tag{9}$$

This metric is particularly important because it provides a means to measure the "quality" of the set $\mathbf{A}$ in the identification of E_k and can be computed a priori, since it is independent of the data set. If the Validity of $\mathbf{A}$ is high, we may trust the answer that occurs by the evaluation of the properties in $\mathbf{A}$. Partial Validity is also bounded by the the total Validity of the definition, $\mathcal{V}(E_k/\, \mathbf{A}) \leq \mathcal{V}(E_k/\, \mathbf{A}_t) \equiv \mathcal{V}(E_k)$.

Finally, we define *Partial Complexity*

$$\mathcal{C}(E_k/\mathbf{A}) \triangleq \sum_{i \in \mathbf{A}} c(t_i) \tag{10}$$

and since the complexity values $c(t_i)$ are nonnegative, $\mathcal{C}(E_k/\mathbf{A}) \leq \mathcal{C}(E_k)$.

D. Semantic Identification Design

Assume that a Complexity threshold C_T is given for the identification of a Semantic Entity using the detailed definition E_k. From the power set $2^{\mathbf{S}_{E_k}}$ of $\mathbf{S}_{E_k}$, we select those subsets $\mathbf{A}_i$ that satisfy the Complexity criterion

$$\mathcal{C}(E_k/\, \underset{i}{\mathbf{A}}) \leq C_T \tag{11}$$

The optimal subset $\mathbf{A}_T^*$ is the one that has the greatest Validity,

$$\mathcal{V}(E_k/\mathbf{A}_T^*) = \max(\mathcal{V}(E_k/\mathbf{A}_i)) \tag{12}$$

Of course, setting the Complexity threshold $C_T \geq \mathcal{C}(E_k)$ would lead to complete evaluation of the definition.

In a similar manner, we may design the identification in terms of Validity if a Validity threshold V_T is given, under which no answer is accepted. Hence, we select the subsets $\mathbf{A}_i$ of $2^{\mathbf{S}_{E_k}}$ that satisfy the Validity criterion

$$\mathcal{V}(E_k/\mathbf{A}_i) \geq V_T \tag{13}$$

In this case, the optimal subset $\mathbf{A}_T^*$ is the one that has the minimum Complexity, i.e.

$$\mathcal{C}(E_k/\mathbf{A}_i) = \min(\mathcal{C}(E_k/\mathbf{A}_i)) \tag{14}$$

Note that a Validity threshold $V_T > \mathcal{V}(E_k)$ cannot be exceeded, so in this case no subset $\mathbf{A}_i$ exists.

One may argue that the problem of finding the optimal subset is itself computationally inefficient, since for n Syntactic Entities the number of subsets is 2^n. However, by modeling this problem as the so-called “Knapsack” [5] problem with a nonlinear gain function (Validity) the problem is solved in pseudo-polynomial time very efficiently. The analysis of the algorithm that solves the optimal subset selection problem is beyond the scope of this work and is the core subject of paper [6].

Apart from the design in terms of Complexity or Validity, one may choose to use variations, suitable for the needs of each application. Examples include the use of fuzzy thresholds, or the use of combined criteria (including both Validity and Complexity).

IV. Non-detailed Definitions

The identification process and optimization have been analyzed so far for the case of detailed definitions, where there is a direct relation between the Syntactic Entities and the Semantic Entity that is defined. However, the Encyclopedia may contain definitions of Semantic Entities in terms of other Semantic Entities, i.e. non-detailed definitions. In this section, three methods are presented that deal with non-detailed definitions using three different approaches, useful for various application scenarios.

A. Direct Substitution

In this case, a non-detailed definition is transformed into a detailed one. By gradually substituting each Semantic Entity participating in a definition we obtain a definition “tree” where the leaf nodes of this tree are the Syntactic Entities participating in the definition, either directly or indirectly. Consider again the example of Figure 1. The definition C is detailed, while the Syntactic Entities participating in the definition B are a (both directly and indirectly) and b (indirectly, through C). In the direct substitution method the goal is to obtain appropriate values F'_{aB} and F_{bB} so that B can be transformed into a detailed definition B_d. For this purpose, we use fuzzy intersection for the transition from b to B via C, consequently

$$F'_{bB} = \mathcal{I}(F_{bC}, F_{CB}). \tag{15}$$

There exist two ways that a is related to B, directly with F_{aB} and through C, with $\mathcal{I}(F_{aC}, F_{CB})$. Fuzzy union is used to combine these values and calculate F'_{aB}, hence

$$F'_{aB} = \mathcal{U}(F_{aB}, \mathcal{I}(F_{aC}, F_{CB})). \tag{16}$$

Note that fuzzy union is used whenever multiple paths exist and *at the junction* of these paths. This means that in order to find the relation between the Entity A and a, one would use the detailed definition of B calculated above, and not apply union at top level. Hence,

$$F'_{aA} = \mathcal{U}(F_{aA}, \mathcal{I}(F'_{bB}, F_{BA})), \tag{17}$$

$$F'_{bA} = \mathcal{I}(F'_{bB}, F_{BA}). \tag{18}$$

All of the above can be rewritten in a mathematically convenient form by using fuzzy relations. Consider the general definition $E = F_{E_{d1}E}/E_{d1} + \ldots + F_{E_{dm}E}/E_{dm} + F_{Y_1E}/Y_1 + \ldots + F_{Y_nE}/Y_n$ where $E_{d1}, \ldots, E_{dm}$ are Semantic Entities with detailed definitions. This definition can be represented using a fuzzy relation $\mathbf{R}_E$, as stated in section III-B. We can obtain a fuzzy relation $\mathbf{R}^c_E$ that provides the weights F for all the Syntactic Entities participating in E either directly or indirectly, i.e. it provides the values F for every element in $\mathbf{S}^c_E = \mathbf{S}_{E_{d1}} \cup \ldots \cup \mathbf{S}_{E_{dm}}$, by

$$\mathbf{R}^c_E = (\mathbf{R}_{E_1} \cup \ldots \cup \mathbf{R}_{E_m}) \circ \mathbf{R}_E \tag{19}$$

where “$\cup$” is the standard union and “$\circ$” is the composition defined in section III-B. By applying this procedure recursively (if any of the definitions E_{di} is not detailed), we obtain a relation providing the weights for all Syntactic Entities and (7) may be used for the computation of Certainty μ_E. In summary, the direct substitution method uses

$$\mathbf{Z} = \mathbf{X} \circ ((\bigcup_{i \in \mathbf{S}^c_E \cap \mathbf{E}} \mathbf{R}^c_i) \circ \mathbf{R}_E) \tag{20}$$

where $\mathbf{R}^c_i$ are the “composite” relations for all non-detailed definitions and the Certainty μ_E is included in $\mathbf{Z}$.

While it is common to use the $\sup$-t composition and the transitive closure of a fuzzy relation $\mathbf{R}$ to find the degree of relationship between the elements of $\mathbf{R}$ (in [7] for example, this approach was used for semantic query expansion) this section proposes a different method to this end. This is because the standard union is not suitable for the design of the identification, since for each subset Validity is determined by the Syntactic Entity with the higher weight F, providing no granularity. On the contrary, for any other fuzzy union operator $\mathcal{U}$ and $F_a > F_b$, we have $\mathcal{U}(F_a, F_b) > F_a$, i.e. the inclusion of b, increases the Validity of the identification. The drawback of this approach stems from the fact that $\mathcal{U}(a, a) > a$ in general, so the transitive closure [7] is not the appropriate choice for the generation of detailed definitions (because the weights already provided would be modified). On the contrary, the procedure presented above, yields the desired results.

B. Subcontractors

In the direct method only the information provided by the Encyclopedia is used for the identification. However it is common for multimedia documents to be pre-annotated in real life applications, meaning that information about certain Entities may already exist and should be used. The subcontractor method assumes that someone else has gone to the trouble of identifying the Semantic Entities involved in a definition (hence their Certainty is available in a document). We may therefore consider these Semantic Entities as Syntactic ones, since information about their existence in a document is already available. Therefore a non-detailed definition may be treated as a detailed one and the computational cost of the evaluation of lower level Semantic Entities is zero.

The idea of the subcontractor actually applies to any method. By knowing the Certainty μ_{E_k} of an Entity E_k we may treat that Entity as a Syntactic one, even if no information is available for other Semantic Entities participating in its definition. The difference is that in Equation (7) $\mathbf{X}$ contains Certainty values for Semantic Entities as well.

C. The Hybrid Method

The hybrid method differs from the direct and subcontracted methods in that it doesn't aim at providing detailed definitions from non-detailed ones, but it is useful in the identification of multiple Semantic Entities in a document. We assume that a pool of algorithms exists, that can be used to evaluate the corresponding Syntactic Entities. Evaluation of a Syntactic Entity Y_i yields a partial Certainty μ_{E_k} for all the Semantic Entities where $Y_i \in \mathbf{S}_{E_k}$. Additionally, each μ_{E_k} adds to the partial Certainty of other Entities μ_{E_l} that use E_k (as in the subcontractor approach) and so on.

Assume a Semantic Entity E described by lower level Semantic Entities with detailed definitions. We calculate μ_E using

$$\mathbf{Z} = (\bigcup_{i \in \mathbf{S}_E \cap \mathbf{E}} \mathbf{X} \circ \mathbf{R}_i^d) \circ \mathbf{R}_E \tag{21}$$

where $\mathbf{R}_i^d$ are relations corresponding to detailed definitions. For Semantic Entities described by E, $\mathbf{Z}$ is used instead of $\mathbf{X}$ and E is treated as a Syntactic Entity.

Hence by executing a set of algorithms in a document, we obtain Certainty values for a large set of Entities. This approach can be useful in the annotation of a document by finding which Entities exist in it and up to what degree, contrary to the case when a single Semantic Entity is identified. Note, however, that this approach is not equivalent to the direct approach. More specifically, transformation of a definition to a detailed one and evaluation of all Syntactic Entities participating in that definition doesn't yield the same Certainty as the hybrid method that recursively computes the Certainty for all the lower level Semantic Entities of the definition. This happens because the distributivity property does not hold for fuzzy unions and intersections in general. An exception to this is the standard fuzzy union, although this operator is not efficient to use with the identification design as mentioned in section IV-A.

V. EXPERIMENTS

A. A Completely Useless but Rather Didactic Example

Mr. Butter loves butterflies, so he has installed a camera in his garden and wishes to be notified whenever a butterfly happens to be passing by. An expert provided a definition for *butterflies*: They have a *shape* that matches a set of sketches, their *flight* has a periodic up and down motion with frequency within a particular range $[f_0, f_1]$ and that they have a tendency to approach and *stop on flowers*. Let the aggregated distance of butterfly shape from a prototype sketch be denoted by d and its velocity by v. On the other hand, *flowers* are identified by their *vivid color* (Mr. Butter's garden has no other vivid colored objects) and also by their shape that should be close to a prototype flower sketch; let d_f denote the distance of this shape from the prototype. After reading section II-B of this paper Mr. Butter organized the provided definitions as follows.

$$B = \text{“Stop on Flower”}/0.5 + \text{“Shape”}/0.8 + \text{“Flight”}/0.9 \tag{22}$$

$$\text{“Stop on Flower”} = \text{“Flower”}/0.9 + (v \text{ close to } 0)/0.8 \tag{23}$$

$$\text{“Shape”} = (\text{small } d)/0.8 \tag{24}$$

$$\text{“Flight”} = (f \text{ close to the range } [f_0, f_1])/0.95 \tag{25}$$

$$\text{“Flower”} = (\text{vivid colors})/0.9 + (\text{small } d_f)/0.6 \tag{26}$$

The Complexity values associated with the Syntactic Entities of the definitions are given in table I.

Syntactic Entity	Complexity (MFLOPS)
v close to 0	200
small d	300
f close to the range $[f_0, f_1]$	400
vivid colors	100
small d_f	300
Total Complexity	1300

TABLE I
COMPLEXITY VALUES FOR THE SYNTACTIC ENTITIES OF B.

Some adjustments with the camera and the equipment in general were made so the method was actually good at identifying butterflies during testing. However, when Mr. Butter tried to use this method on his own rather old computer he found that the method required more computational resources than it could handle. By skipping section III-D he thought that the ideas of section IV-B could simplify the process if the subcontractor approach was used for the Semantic Entity "Flower", since this Entity can be identified once every morning and be used for the rest of the day. The gain in complexity was $c(\text{small } d_f) + c(\text{vivid colors}) = 400$ and $C_{total} = 900$. Nevertheless, the Complexity was still too high. It was then that Mr. Butter decided to revisit section III-D and apply the proposed design method.

Mr. Butter adopted the direct approach (section IV-A), so the definition (22) was transformed into a detailed definition B'. As fuzzy intersection and union operators he used the algebraic product and sum respectively ($\mathcal{I}(a,b) = ab$, $\mathcal{U}(a,b) = a + b - ab$). The result was

$$\begin{aligned} B' = & (v \text{ close to } 0)/0.4 + (\text{small } d)/0.64 \\ & + (f \text{ close to the range } [f_0, f_1])/0.855 \\ & + \text{“Flower”}/0.45. \end{aligned} \tag{27}$$

The total Validity and Complexity of the definition are $V_t = 0.98277$ and $C_t = 900$. By setting a threshold

$C_T = 400$ and designing in terms of Complexity, the resulting Validity is $V^* = 0.92025$, that occurs by evaluating (f close to the range $[f_0, f_1]$), and taking into account the Semantic Entity "Flower" provided by the subcontractor method. The Complexity of the identification is $C = C_T = 400$ and Mr. Butter is happy now.

This is a rather extreme case, where we assumed that the Syntactic Entity (f close to the range $[f_0, f_1]$) is much more important in the identification than the others, so the increase in Validity by employing this Entity is high. Moreover, Validity increases by the use of a subcontractor for the Entity "Flower" that is precomputed once for the scene (and which is always employed, since it has no Complexity cost). It is important to notice that the design method's results are useful only if the values of the weights F correspond to the importance of the Entities in the identification as well as the accuracy and robustness of the algorithms employed to perform the evaluation. If this is not the case, the Validity values obtained do not correspond to the accuracy of the actual results: High Validity essentially means that high Certainty will occur during the identification if a butterfly actually exists and lower whenever it doesn't.

B. Evaluation Using Random Values

Seriously now, we may examine the value of the proposed method by using detailed definitions with randomly distributed values of Complexity and weights F. Employing only a subset of the Syntactic Entities in the definitions leads to a reduction in both Validity and Complexity. However, selecting the optimal subset minimizes the Validity loss, while Complexity satisfies the requirements posed by a threshold C. Two definitions, consisting of 30 Syntactic Entities each, were used and Figure 2 demonstrates the results of design in terms of Complexity for Various thresholds. Complexities and weights F have random values obtained from uniform distributions for the first definition, while the second definition uses normally distributed values. Note that for a threshold $C_T = 12$ the Validity is $V \approx 0.9$ while the total Complexity of the definition is $C_t = 132$ for the uniformly distributed values. Furthermore this experiment along with similar experiments that have been conducted show that the method is very efficient when dealing with widely distributed values (their variance is high), contrary to the case of distributions with similar values. This can be justified by the fact that Syntactic Entities with similar values of Complexity and weights F make the selection between them unimportant, since they can be considered "equivalent" under our scope.

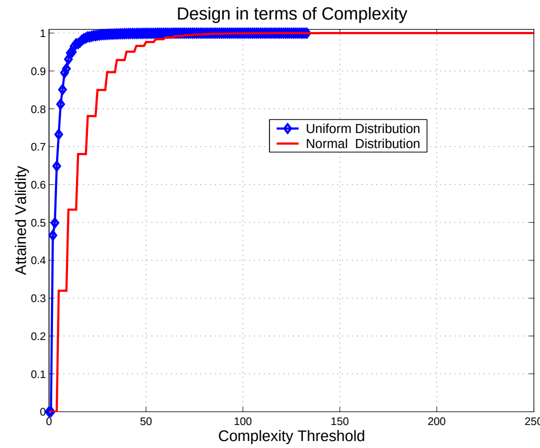

Fig. 2. Design in terms of Complexity.

VI. Discussion

A general method for modeling the semantic identification process has been presented. Each feature that is employed for the identification is assigned a complexity value and a weight that denotes its importance in the identification. This allows the design of the identification taking into account possible limitations in computational complexity or requirements on the Validity of the identification. The major advantage of this work is that this model is flexible enough to be used by semantic identification techniques available in the relative literature.

Future plans include the extension of the Fuzzy Semantic Encyclopedia so as to be able to use more complex and expressive mathematical logics, such as description logics (see [8] and [9]) and the evaluation of the method in dynamic, real time environments, where the design of the identification is constantly reevaluated during the process. Although in this work we assume that the weights F are predefined in the Encyclopedia, it is clearly a major issue to develop algorithms that compute them, either automatically or semi-automatically, by using appropriate training techniques.

References

[1] B.S. Manjunath, Phillipe Salembier, and Thomas Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, 2002.

[2] R. Leonardi, P. Migliorati, and M. Prandini. Semantic indexing of sport program sequences by audio-visual analysis. In *IEEE International Conference on Image Processing, ICIP*, Barcelona, Spain, September 2003.

[3] J. Assfalg, M. Bertini, C. Colombo, and A. Del Bimbo. Semantic annotation of sports videos. *IEEE Multimedia*, 9(2), April-June 2002.

[4] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic; Theory and Applications*. Prentice Hall, 1995.

[5] Hans Kelleler, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.

[6] M. Falelakis, C. Diou, A. Valsamidis, and A. Delopoulos. Dynamic semantic identifi cation with complexity constraints as a knapsack problem. In *Fuzz-IEEE Conference*, May 2005.

[7] 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.*, 34(2), March 2004.

[8] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[9] Umberto Straccia. Reasoning within fuzzy description logics. *Journal of Artifi cial Intelligence Research*, 14, April 2001.