Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 286 (2016) 1–29

FUZZY sets and systems

www.elsevier.com/locate/fss

Fuzzy constraint satisfaction problem for model-based image interpretation

Maria Carolina Vanegas [a,1], Isabelle Bloch [b,*], Jordi Inglada [c,d]

[a] *Definiens, Germany*
[b] *Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI, Paris, France*
[c] *Centre National d'Etudes Spatiales, Toulouse, France*
[d] *CESBIO UMR 5126, Toulouse, France*

Received 7 August 2013; received in revised form 15 October 2014; accepted 17 October 2014

Available online 4 November 2014

Abstract

Scene interpretation guided by a generic model benefits from structural representations of objects and their spatial relationships. In this paper, we consider nested conceptual graphs for encoding objects and groups of objects, spatial relationships between objects or between groups of objects, along with the imprecision and uncertainty attached to the formal representations of such relationships. Scene interpretation is then formalized as a graph homomorphism problem for the identification of possibly multiple instances of the model in an image. We propose an extension of fuzzy constraint satisfaction problems (FCSP) to deal with complex objects. In particular, we extend FCSP arc-consistency checking to deal with groups of objects which can be related among them or have a spatial property such as being aligned. The instantiations of the model in the image are obtained by solving a FCSP. This framework is illustrated on the example of interpretation of Earth observation images. A method is proposed to find the instantiations of a nested conceptual graph, representing a generic model of the scene (such as harbor or airport) in an unlabeled image. Experimental results on high resolution satellite images show that the proposed approach successfully recognizes a given spatial configuration and is robust to image segmentation errors. The results demonstrate the interest of using complex spatial relations for the interpretation of images.
© 2014 Elsevier B.V. All rights reserved.

Keywords: Image understanding; Structural model; Fuzzy constraint satisfaction problem; Spatial relations; Earth observation images; Nested conceptual graphs

1. Introduction

Image interpretation consists in recognizing different objects which compose a scene, understanding their spatial organization, and providing a description of this scene and a semantic labeling of the image. To interpret an image it

* Corresponding author.
E-mail addresses: cvanegas@definiens.com (M.C. Vanegas), isabelle.bloch@telecom-paristech.fr (I. Bloch), jordi.inglada@cesbio.cnes.fr (J. Inglada).

[1] This work was performed during M.C. Vanegas's PhD thesis at Telecom ParisTech and CNES.

http://dx.doi.org/10.1016/j.fss.2014.10.025
0165-0114/© 2014 Elsevier B.V. All rights reserved.

is necessary to use information within the image, but also contextual information relevant to the interpretation task. The contextual information should allow us to answer questions such as: which are the objects of interest? How to identify these objects or their parts? And how are these objects or parts related? This information depends on the domain, the objective of the description and the application. Thus when performing an automatic image interpretation a knowledge representation system should be developed. In this system we should be capable of encoding information about objects, spatial relations between objects or object parts, extraction algorithms, etc.

In this paper we focus on the interpretation of *very high resolution Earth observation images*, as an illustration. These images contain a large amount of information. They are the outcome of the combination of many different intensities that can represent natural concepts such as vegetation, geomorphological and hydrological concepts, man-made objects such as buildings and roads, and artifacts caused by variations in illumination of the terrain, such as shadows [43]. This large quantity of information allows the description of the images at different conceptual levels [23]:

- individual objects, for instance a house, a tree, a road segment,
- land cover type, for instance water, bare land, vegetation,
- complex or composite objects, which are new semantic objects formed by several spatially related individual objects, for instance airports, harbors, train stations, nuclear power plants, toll gates, stadiums, etc.

The conceptual level used to describe the image depends on the objective of the description as well as the resolution of the image. Moreover, Earth observation images contain objects of different sizes, which makes it unfeasible to analyze all the concepts of the image at the same scale. For instance, a building can be identified at a very high resolution using its shape, a city is better identified at a lower resolution as a texture. Therefore, according to the level of concept we are interested in, we should choose an appropriate scale of observation. In this work we concentrate on the interpretation of complex objects. One of the difficulties of this task lies in determining the important details to extract the objects which compose complex objects. Objects belonging to the same complex object can be observable at different scales. Furthermore, some objects in a complex scene cannot always be recognized with traditional methods using spectral and textural features. They often require the recognition of other objects having a spatial relation with them, and then use the spatial relation to identify them. Hence the spatial relations and the spatial arrangements of objects and of complex objects and their parts in the scene are of prime importance for the recognition of complex objects. Another difficulty encountered in the interpretation of complex objects is that the spatial arrangements can be expressed as **complex relationships**, i.e. spatial relations that are not necessarily expressed between two objects but between a group of objects and an object, or groups of objects having a spatial property, such as alignment. Hence, the interpretation method should be able to encode these complex relationships and deal with them.

In this work we propose a method to identify complex objects in an Earth observation image. We make use of knowledge on the spatial structure of the complex objects to guide their identification in the image. We describe a complex object through a nested conceptual graph. This model allows the representation of spatial information of the complex object, including complex spatial relationships. We assume that the image is already segmented and we only concentrate on the problem of labeling the regions according to the knowledge and information supplied by the model.

We formulate the problem of labeling the image regions as a homomorphism from the model to the image regions. Modeling the problem as a homomorphism permits to map the model to several regions of the image (several instantiations), and thus account for uncertainty within the model. By uncertainty in the model we mean that even if all the objects in the model appear in the image, we are not certain about the number of instantiations. For instance in a harbor, a dock can have none, one or several boats adjacent to it. This type of uncertainty is present in Earth observation images. Moreover, when a model is used to represent the spatial arrangement of the objects that should appear in the scene, other types of information imperfections can be present:

- Imprecision on spatial relations. Many spatial relations can be imprecise by nature. Their satisfaction depends on the context or even on the size of the objects. For instance the relation *near* has a different interpretation according to the size of the reference object: the distance used to determine whether a building is near an airport is different from the one used to determine whether a building is near a tree.
- Uncertainty with labeling objects in the image. When labeling the objects in the image after a segmentation, labels may be uncertain.

- Imprecision on the objects in the image. This can be due to different causes, such as the discretization of space (passing from a continuous scene to a digital image), or the processing methods (e.g. segmentation).

We express the problem of finding a graph homomorphism as a *Fuzzy Constraint Satisfaction Problem* (FCSP). Using a FCSP allows us to consider the different sources of imperfections. The imprecision of spatial relations is dealt by modeling the spatial relations as fuzzy relations which are handled by FCSP. The other two types of information imperfections are modeled by using fuzzy membership functions (that consider information from the model) to perform an initial labeling of the objects in the image.

The main two contributions of this paper are (i) an interpretation method that starts from an unlabeled segmented image (possibly imprecisely and inaccurately segmented) and exploits knowledge about the scene to be recognized, and (ii) translate the problem as a FCSP and adapt the FCSP formalism to deal with complex spatial relations. The proposed framework was developed in [44].

We formulate the problem of image interpretation within the framework of Knowledge Based Systems (KBS). In Section 2, we introduce KBS, their use in image interpretation, and discuss the choice of the proposed method. In Section 3, we discuss nested conceptual graphs and their representation capabilities. In Section 4 we explain how the proposed problem can be represented as a Fuzzy CSP. The FCSP formalism has to be extended in order to allow the representation of embedded information, which is done in Section 5. Finally in Section 6, we illustrate the proposed method on two examples.

2. Knowledge Based Systems (KBS) for image interpretation

The important role played by knowledge in image interpretation explains the large development of KBS's in this domain. A review of such systems can be found in [13,27,43]. KBS's are inspired by human reasoning, and consist in representing and modeling the knowledge relative to a domain. Their objective is to reason on this knowledge in order to solve a concrete problem, such as identification, recognition, classification, diagnosis, configuration and planning, among others [27]. These systems are usually composed of three parts: the knowledge base, the observation base and the reasoning components.

For image interpretation, the knowledge base is typically composed of three types of knowledge:

- Image processing knowledge: it is used to extract low level features from the image and their numerical descriptions, so that they can help identifying objects of interest in the image.
- Domain knowledge: it concerns knowledge about the semantics of the domain of the image.
- Knowledge about the mapping between image features and concepts: it establishes the link between the two previous types of knowledge. It concerns the knowledge used to map low level features and high level concepts related to the domain of interest. This mapping problem is also known as the *semantic gap* [26,41].

Image interpretation requires reasoning strategies which can deal with information imperfections. In the following, we review some methods for reasoning under uncertainty in the spatial domain. We discuss the reasoning aspects and how imperfections are represented.

Image interpretation methods can be divided into two strategies. In the first one, an image segmentation is performed followed by a labeling, i.e. mapping image regions into concepts of a model. In the second one, the segmentation and interpretation are performed simultaneously.

2.1. Segmentation followed by labeling

When following this strategy one should consider uncertainty when assigning a label to an image region. In [24,29,35,39,47] an initial labeling of the regions is performed, and spatial relations are used to refine this labeling or to extract the objects of interest. For instance in [39] an initial membership degree is assigned to each region, representing a specific concept according to a classification score. The relabeling is improved by considering the binary spatial relations between concepts. Each relation is modeled as a fuzzy spatial relation. The spatial information is injected into a fuzzy constraint satisfaction problem, where the spatial relations represent the constraints, the concepts, the variables, and the domain of the variables is constituted by the image regions. This approach assumes that the initial segmentation is correct, which is very restrictive because usually a generic segmentation does not allow discriminating between objects belonging to different semantic concepts. For instance, the roof of a building is more homogeneous than a forest when observed in a satellite image. Thus the object of interest can be divided into several regions or two objects can be included in the same region. To overcome the problems raised by a generic segmentation it is possible to: (i) include high level knowledge in the segmentation, (ii) use a multi-scale segmentation or (iii) consider an over-segmentation of the image where a concept of the model is represented by several regions. The first possibility is discussed in the next section, describing joint image segmentation and mapping systems. The multi-scale segmentation approach is widely used in OBIA (Object Based Image Analysis) methods [2,29,33,40]. The main issue is to label relevant regions and to discard those not corresponding to objects, by exploiting the different scales. This labeling may rely on object properties and crisp spatial relations using rules as in [29], or neighboring regions as in [24]. These approaches illustrate the importance of considering spatial relations and uncertainty.

Finally the last possibility to get around the problems of generic segmentation is to perform an over-segmentation of the image. By performing an over-segmentation, the correspondence between the model and the regions of the image is not one to one, but a group of regions can represent an instantiation of a concept. This approach was studied in [15,16,37].

2.2. Joint image segmentation and mapping

In [6,9,34] the segmentation and interpretation problems are addressed simultaneously, and the information from the spatial relations is directly used to help the segmentation process. The authors applied this approach to the interpretation of medical images, which can have a high variability. In [6,9], the interpretation starts with the segmentation of an anatomical structure which is relatively easy to identify. For instance in magnetic resonance images of the brain the right and left lateral ventricules are easy to segment due to their high contrast with the neighboring structures. Then the method searches another anatomical structure using its geometric properties and the spatial relations with the previously recognized structures. This method is performed sequentially. A strategy to optimize the order of recognition is proposed in [21]. In [34], the problem is expressed as a Constraint Satisfaction Network, encoding anatomical structures and relations. Propagators are defined for each relation, and iteratively applied to reduce the domains of each variable (i.e. anatomical structure to be recognized). This global approach provides upper and lower bounds of each structure, which are close to the desired result, and in which a precise segmentation is then performed.

The methods following the strategy of jointly segmenting and interpreting the image are innovative in the sense that they use spatial relations as another source of information to guide the segmentation and not only to verify it. Moreover, they account for imprecision of spatial relations as well as imprecision of objects in the image.

2.3. Discussion and proposed framework

When dealing with Earth observation images, the two strategies described above have pros and cons, that led us to propose an intermediate solution. Two strong characteristics of several model-based methods are the use of spatial relations between the objects and the modeling of different types of information imperfections. They will be also characteristics of the proposed approach.

The methods which perform jointly segmentation and interpretation are not directly confronted with the problem of correctly labeling a region. Moreover, these methods yield better results than the ones which separate the problem into segmentation followed by interpretation, because all the spatial information as well as the geometry and intensity are considered for making the decision, i.e. segment and recognize a structure. These types of methods have been applied for instance to the interpretation of brain and thorax images, which are strongly structured scenes and for which prior knowledge about the expected objects is available. Hence, it is possible to construct models containing all the structures that appear in the scene (the case of pathologies can be included in the model too [1]), as well as the relations between them, and therefore allow a very constrained formulation of the problem.

As mentioned in Section 1, in Earth observation images there is uncertainty with respect to the model. Thus if we try to formulate the interpretation problem using a similar approach as in [6,9,34], it may happen that, for a given object in the model, the region of space which represents the conjunction of the relations that should be satisfied by this object, according to the model, is not sufficiently restricted to contribute to the segmentation of the object. Additionally, multiple instances are difficult to handle.

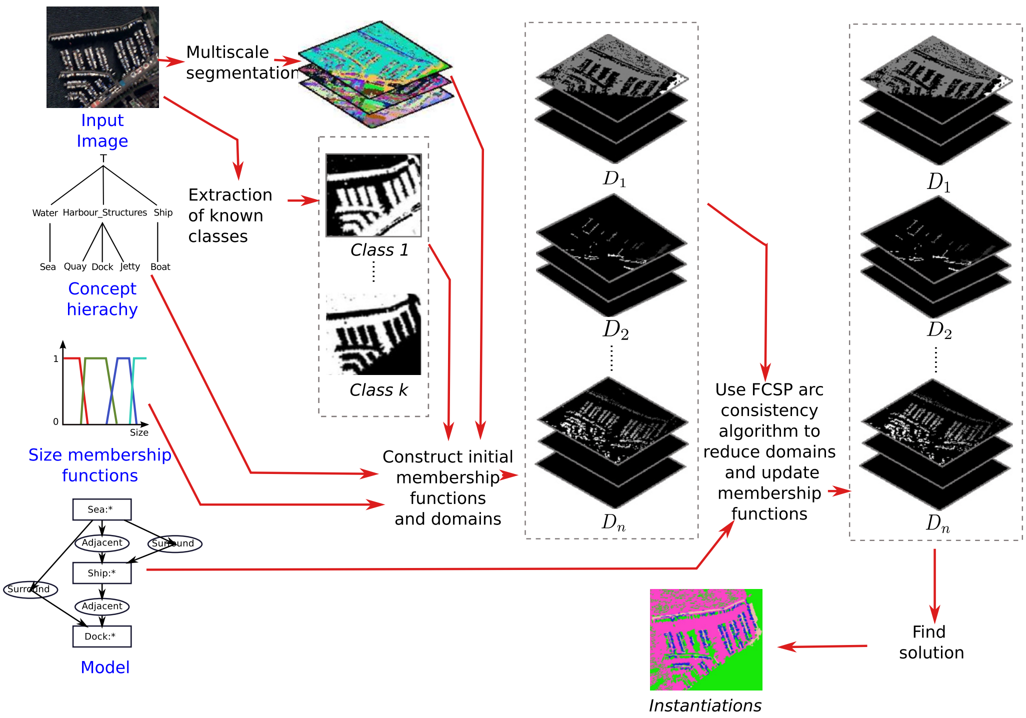


Fig. 1. Proposed method for determining the model's instantiations.

The methods which perform an over-segmentation of the image, and use spatial reasoning to bring together regions to form objects, also need a very constrained formulation of the model. Furthermore they usually have a high computational cost.

We propose an intermediate solution which combines several interesting characteristics of previous approaches. Fig. 1 illustrates the main ideas of our proposed method. First a multi-scale segmentation provides candidates for objects in the scene. Several methods can be used for this step, which is out of the scope of this paper. Since none of them provides perfect results, we define membership functions over the set of regions for each object in the model, as in [39]. These membership functions are designed using information from a classification procedure or knowledge about the radiometry of objects in the images. Using fuzzy spatial relations we identify the different concepts in the model. The problem is formalized as a Fuzzy Constraint Satisfaction Problem, which is the main contribution of this paper. We use an arc-consistency algorithm to reduce the number of possible regions that can be part of an instantiation of the model, and finally we find the instantiations.

3. Structural model

The model used to represent the scene has to be flexible enough in order to allow the representation of the following spatial situations which are frequent in Earth observation images:

S1. Two or more objects satisfying a spatial relation. For example, a "fountain between a road and a house". Note that spatial relations can have any arity.

S2. A group of objects satisfying a spatial relation with another object, with potential relations among the members of the group. This allows the representation of situations such as "a house adjacent to a road", and "the house has a shadow".
In this situation it is necessary to consider the house and the shadow as a group because we do not know in advance whether the shadow can fall onto the road or not. Thus if we consider the house and the shadow as a group we only need to specify that the group is adjacent to the road without worrying about the exact relations

between the house, its shadow and the road (i.e. the house and the road may appear as adjacent or not in the image depending on the position of the shadow).

S3. A group of objects, as in situation S2, but the group has also a spatial property which characterizes the group, for example, a road parallel to a group of aligned trees or a group of houses surrounding a park. In this case the difficulty comes from the fact that the number of objects is unknown.

3.1. Conceptual graphs and nested conceptual graphs

We propose to build the model on the notion of conceptual graph, to represent adequately the situations S1–S3 described above. Conceptual graphs, introduced in [42], allow us to represent the first type of spatial situation (S1). Conceptual graphs are built over a vocabulary $\mathcal{V} = \{(T_C, \leq_C), (T_R, \leq_R), I\}$, where T_C and T_R correspond to the ontologies representing the set of concepts and relations of the domain, and $\leq_C$ and $\leq_R$ their respective ordering relation. The set I, which corresponds to a set of names, called individual markers, is used to denote specific objects or entities.

Definition 1 *(Conceptual Graph (CG)).* (See [7].) A conceptual graph is a bipartite graph denoted by $G = \{\tilde{\mathcal{N}}_C, \mathcal{N}_R, \mathcal{E}, \tilde{l}\}$ where:

- $\tilde{\mathcal{N}}_C$ and $\mathcal{N}_R$ are the concept node and the relation node sets, respectively. The set of nodes of G is equal to $\tilde{\mathcal{N}}_C \cup \mathcal{N}_R$,
- $\mathcal{E}$ is the family of edges,
- $\tilde{l}$ is a labeling function of the nodes and edges of G which satisfies:
  - a concept node $c \in \tilde{\mathcal{N}}_C$ is labeled by $\tilde{l}(c) = (\text{type}(c), \text{marker}(c))$, where $\text{type}(c) \in T_C$, $\text{marker}(c) \in I \cup \{*\}$, and $*$ denotes a generic marker;
  - a relation node $r \in \mathcal{N}_R$ is labeled by $\tilde{l}(r) \in T_R$. $\tilde{l}(r)$ is also called the type of r and is denoted by $\text{type}(r)$;
  - the degree of a relation node r is equal to the arity of $\text{type}(r)$;
  - edges incident to a relation node r are totally ordered and they are labeled from 1 to $\text{arity}(\text{type}(r))$.

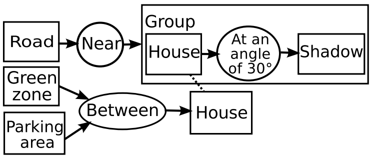
This definition of conceptual graphs allows us to represent relations of any arity between the concept nodes, and thus the spatial situation S1. The set I is used to represent specific instantiations of the concepts. However, if we do not want to specify a particular instantiation, then it is possible to use the generic marker $*$. These types of graphs are appropriate to represent spatial relations between objects. However, they cannot represent hierarchically structured knowledge.

In [42] conceptual graphs were extended to *nested conceptual graphs* to allow the representation of internal and external information, zooming, partial description of an entity, or specific contexts. In a nested concept graph, the concept nodes can contain a conceptual graph. This formalism allows the representation of the spatial situations S2 and S3.

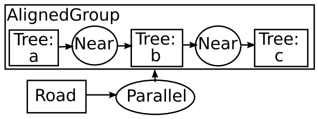
We will refer to nodes which contain a conceptual graph as complex concept nodes. To specify that a node is a complex concept node, a third field is added to each conceptual node, called *description*. Concept nodes which are not complex will have an empty *description* field, denoted by $**$.

Definition 2 *(Nested Conceptual Graph (NCG)).* A nested conceptual graph is a bipartite graph denoted by $G = \{\mathcal{N}_C, \mathcal{N}_R, \mathcal{E}, l\}$ where:

- $\mathcal{N}_C$ and $\mathcal{N}_R$ are the concept node and the relation node sets, respectively. The set of nodes of G is equal to $\mathcal{N}_C \cup \mathcal{N}_R$,
- $\mathcal{E}$ is the family of edges,
- l is a labeling function of the nodes and edges of G which satisfies:
  - a concept node $c \in \mathcal{N}_C$ is labeled by $l(c) = (\text{type}(c), \text{marker}(c), \text{description}(c))$, where $\text{type}(c) \in T_C$, $\text{marker}(c) \in I \cup \{*\}$ and $\text{description}(c) \in \{**\} \cup Desc$, where $Desc$ is a set containing the labels of the descriptions;
  - a relation node $r \in \mathcal{N}_R$ is labeled by $l(r) \in T_R$;
  - edges incident to a relation node r are totally ordered and they are labeled from 1 to $\text{arity}(\text{type}(r))$.



(a) Nested conceptual graph with coreference links, represented as dotted lines.



(b) Nested conceptual graph representing a group of aligned trees, which are parallel to a road. The description "Aligned Group" represents the spatial arrangement of the group.

Fig. 2. Examples of nested conceptual graphs. Concept nodes are represented by rectangles and relation nodes by ellipses.

The set of complex concept nodes of a conceptual graph is denoted by $D(G)$. The nodes inside a complex concept node are called child nodes. A nested conceptual graph can be recursively defined from a basic conceptual graph (Definition 1) by adding the description field to the labeling of the concept nodes. Another representation for nested conceptual graphs is a tree of basic conceptual graphs (see [7] for more details). The label of a simple node c is written as type(c) : marker(c), and for simplicity, when the marker of a node is the generic marker $*$ then we just label it as type(c).

To represent relations between objects inside a complex concept node and concept nodes outside it, we can use a coreference concept. Coreference concepts represent two concepts which are equivalent and represent the same entity, and they are joined by a coreference link. The use of coreference concepts is important since the knowledge inside the complex node is contextualized by the hierarchical structure representing the group. For instance, Fig. 2(a) represents a house with its shadow as a group (the group is drawn as a box) which is near a road. The additional information that the house is between a green zone and a parking area is encoded by a coreference link (drawn as a dotted line).

We represent the groups endowed with a spatial property (spatial situation S3), as a complex concept node with the description explaining the spatial property. For instance, Fig. 2(b) represents a group of trees where the trees are arranged in a line, and we represent this property by adding the description "Aligned group" to the complex concept node. In the NCG the group is parallel to a road. To represent that there are different elements (different instances of tree) we use a different marker to label each element. In [46] the spatial situation of objects being arranged in a line is studied and a measure to determine whether a group satisfies this property is proposed. For simplicity we will refer to the objects arranged in a line as "group of aligned objects". Moreover, in this complex concept node we also represent the distance relation between the neighboring objects of the group.

3.2. Conceptual graph homomorphism

Nested conceptual graphs are not only appropriate to represent the spatial information used to describe the image, but also their reasoning mechanism is appropriate for dealing with the characteristics of Earth observation images. Reasoning in concept graphs is usually done through graph homomorphism. A conceptual graph homomorphism is defined in the following way.

Definition 3 *(Conceptual graph homomorphism).* (See [7].) Let $G_T = (\mathcal{N}_{C_T}, \mathcal{N}_{R_T}, \mathcal{E}_T, l_T)$ and $G_H = (\mathcal{N}_{C_H}, \mathcal{N}_{R_H}, \mathcal{E}_H, l_H)$ be two conceptual graphs defined over the same vocabulary $\mathcal{V} = \{(T_C, \leq_C), (T_R, \leq_R), I\}$. A homomorphism ϕ from G_T to G_H is a mapping from $\mathcal{N}_{C_T} \cup \mathcal{N}_{R_T}$ to $\mathcal{N}_{C_H} \cup \mathcal{N}_{R_H}$, which satisfies:

- $\forall (r, i, c) \in \mathcal{E}_T,\ (\phi(r), i, \phi(c)) \in \mathcal{E}_H$,
- $\forall e_C \in \mathcal{N}_{C_T}, l_H(\phi(e_C)) \leq_C l_T(e_C)$, and
- $\forall e_R \in \mathcal{N}_{C_R}, l_H(\phi(e_R)) \leq_R l_T(e_R)$,

where $(r, i, c) \in \mathcal{E}$ represents the edge labeled i between a relation r and a concept c, which means that c is the i-th argument of r. This definition implies that the homomorphism preserves the edges and may decrease concept and relation labels according to $\leq_C$ and $\leq_R$, respectively.

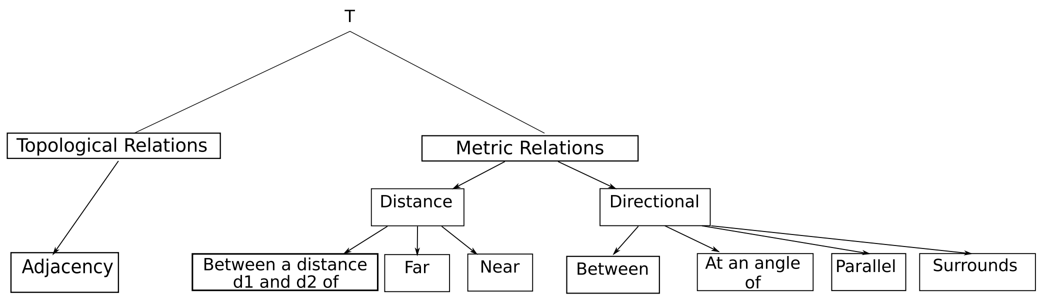


Fig. 3. Excerpt of the spatial relation hierarchy [25].

Using graph homomorphisms allows one graph model to be mapped to several instantiations in the image. Furthermore, we can represent a group of aligned objects, where the number of objects in the alignment is unknown, as a complex concept node with only three concept objects, because there is a homomorphism between an aligned group of objects with three elements and an aligned group containing more elements.

To find a graph homomorphism there are several algorithmic possibilities (see [7] for a review). One possibility is to express the problem as a Constraint Satisfaction Problem (CSP), where the relations represent the constraints, the concept nodes represent the variables of the CSP, and the values of the CSP are the regions of the image. Here we propose to use Fuzzy Constraint Satisfaction Problems (FCSP) to account for the information imperfections that we want to consider in our problem. FCSPs are introduced in Section 4.

3.3. Vocabulary

The vocabulary is composed of three parts: concept, relation hierarchies and individual marker sets. In our application the concept hierarchy depends on the scene that we want to interpret, as it contains the objects that are found in the scene and which we want to recognize.

The relation hierarchy is based on the ontology proposed in [25] (Fig. 3). All these relations have been modeled as fuzzy spatial relations in [4,5,45,46]. Fuzzy representations allow us to fill the semantic gap between conceptual representations and the parameter or image domain, and the semantics of the relations are defined for each domain (typically, the parameters of the membership functions are tuned or learned according to the application, leading to different semantics of "near", for instance depending on whether anatomical structures in medical images or objects in Earth observation images are considered). A relation is modeled:

- either as a fuzzy landscape, which defines a fuzzy region of space where the relation is satisfied with respect to a reference object,
- or as a degree of satisfaction, that allows us to determine to which degree the relation is satisfied between two given objects.

In the first case, the degree of satisfaction of a relation between a target object and the reference object is obtained by measuring the degree to which the target object is included in the region defined by the relation [4,5]. In this work we use an average measure. For instance, suppose that a and b are two regions of the image, which represent two objects, and we want to evaluate to which degree "a is near b". Let γ^b_{near} be the membership function representing the fuzzy landscape defining the region "near b", then the degree of satisfaction of the relation "a is near b" is given by the average value of the membership function γ^b_{near} over all the pixels of the region representing the object a:

$$\mu_{\text{near}}(b,a) = \frac{\sum_{q\in a} \gamma^b_{\text{near}}(q)}{|a|}. \tag{1}$$

In our experiments, all the metric relations except the parallel relation are modeled as fuzzy landscapes. The parallel relation is modeled as a degree of satisfaction. It is defined as a conjunction of the satisfaction of a fuzzy landscape (as in Eq. (1)) and a fuzzy relation measuring the similarity between the orientations of the two objects [46]. The adjacency relation is also modeled as a degree of satisfaction [4]:

$$\mu_{\text{adjacency}}(a,b) = \mu_{\text{int}}\big(\delta_\nu(a), b\big), \tag{2}$$

where δ_ν is a fuzzy morphological dilation and μ_{int} defines a degree of intersection between two fuzzy sets.

In addition to these relations we also consider the spatial property of alignment which has been modeled as a fuzzy spatial relation in [46]. In [46] a satisfaction degree of the alignment relation for a group of objects is proposed, as well as a method to extract the groups of objects which are aligned to a certain degree. In the proposed model the alignment relation also implies the distance relations between the elements of the group. These distance relations are always specified in the conceptual graph, as in Fig. 2(b), however they are only considered when determining whether a group satisfies the spatial property of being aligned.

4. Fuzzy CSP

In this section, we summarize the main definitions and algorithms of Fuzzy CSP (FCSP). A Constraint Satisfaction Problem (CSP) is a generic framework for representing and solving problems whose aim is to find the solutions to a set of constraints. A constraint represents a relation, and a constraint satisfaction problem states which relations should hold between a set of decision variables. The authors in [17] extended CSP to FCSP in order to deal with flexible constraints. Such constraints include: fuzzy relations, soft constraints which express preferences between relations, and prioritized constraints which express the constraints which can be violated in the case where there exists a conflict. These flexible constraints were first considered in [38] to label objects in a scene.

The FCSP formalism is well adapted to our problem since it allows us to represent fuzzy relations. However, the representation of complex concept nodes needs an adaptation that is presented in Section 5.3, as a new contribution of this paper.

4.1. Definitions

Definition 4 *(Fuzzy CSP).* A FCSP is defined as $\mathcal{P} = \{\mathcal{X}, \mathcal{D}, \mathcal{C}\}$, where:

- $\mathcal{X} = \{x_1, \dots, x_n\}$ is a set of variables;
- $\mathcal{D} = \{D_1, \dots, D_n\}$ is a set of domains. Each domain D_i is associated with a variable x_i, and represents the values which can be assigned to x_i;
- $\mathcal{C} = \{C_1, \dots, C_t\}$ is a set of flexible constraints. A flexible constraint C_k is defined by a pair $\langle R_k, S_k\rangle$, where $S_k \subset \mathcal{X}$ is the set of variables which are involved in C_k, and R_k is a fuzzy relation over the Cartesian product of the domains $D_{k_1} \times \dots \times D_{k_m}$ of the variables in S_k. R_k is defined through its membership function μ_{R_k} : $D_{k_1} \times \dots \times D_{k_m} \to [0,1]$. For $V = \{v_{k_1}, \dots, v_{k_m}\} \in D_{k_1} \times \dots \times D_{k_m}$, $\mu_{R_k}(v_{k_1}, \dots, v_{k_m})$ represents the degree to which V satisfies the constraint C_k.

In our case, $\mathcal{X}$ represents the model of the objects that we want to instantiate in the image. They are represented as the concept nodes of the nested conceptual graph. $\mathcal{D}$ corresponds to the regions in the image obtained from a segmentation. The relations R_k represent the relation nodes of the model which are modeled as fuzzy spatial relations.

Given an instantiation $\{v_1, \dots, v_n\} \in D_1 \times \dots \times D_n$, the degree to which $\{v_1, \dots, v_n\}$ satisfies $\mathcal{P}$ is called the consistency degree and is defined as the conjunction of the satisfaction of each of its constraints [17]:

$$\text{cons}(v_1, \dots, v_n) = \min_{C_k \in \mathcal{C}} \mu_{R_k}\big((v_1, \dots, v_n) \downarrow_{S_k}\big) \tag{3}$$

where $(v_1, \dots, v_n) \downarrow_{S_k}$ represents the projection of $(v_1, \dots, v_n)$ onto the set of variables S_k. The degree to which a value $v \in D_i$ is suitable for representing a variable x_i is represented as a fuzzy set μ_{x_i} over D_i:

$$\begin{aligned} \mu_{x_i} : D_i &\longrightarrow [0,1] \\ v &\longmapsto \mu_{x_i}(v) \end{aligned}$$

where μ_{x_i} accounts for the imprecise or incomplete knowledge about x_i. We will refer to μ_{x_i} as the membership function of the variable x_i. Finding the instantiations which satisfy $\mathcal{P}$ to a degree $\alpha \in [0,1]$ is NP-Hard [3]. Thus the problem is usually simplified by applying local consistency algorithms, and then searching for a solution in the reduced problem. A very common type of local consistency is arc-consistency.

Arc-consistency for FCSP containing only binary constraints was defined in [17]. Given a FCSP $\mathcal{P} = \{\mathcal{X}, \mathcal{D}, \mathcal{C}\}$ of fuzzy constraints, $\mathcal{P}$ is said to be arc-consistent if, for every constraint C_k involving the variables x_i and x_j, every $u \in D_i$ satisfies:

$$\mu_{x_i}(u) \leq \sup_{v \in D_j} \min\big(\mu_{R_k}(u, v), \mu_{x_j}(v)\big). \tag{4}$$

This means that the fuzzy set μ_{x_i}, representing the possible values of x_i, should be included in the projection on x_i of the conjunction of the fuzzy set μ_{R_k} with the cylindrical extension of the fuzzy set μ_{x_j}.

For m-ary constraints we use the generalized arc consistency for non-binary constraints as defined in [10] for Valued CSP. Let C_k be a m-ary constraint and x_i a variable belonging to S_k. Then arc-consistency is given by:

$$\mu_{x_i}(v) \leq \sup_{A=(a_{k_1},\ldots,a_{k_m}) \in D_{k_1} \times \ldots \times D_{k_m} : A\downarrow_i = v} \min\Big(\mu_{R_k}(a_{k_1}, \ldots, a_{k_m}), \min_{\substack{k_j \in \{k_1,\ldots,k_m\} \\ j \neq i}} \mu_{x_j}(a_{k_j})\Big). \tag{5}$$

This definition of arc-consistency corresponds to 2-consistency in the Soft CSP framework [3].

4.2. The FAC-3 algorithm

Algorithm 1 shows a generalization of the FAC-3 algorithm to deal with fuzzy constraints of any arity. In the following we refer to this algorithm as FAC-3. This algorithm is a specialization of the algorithm presented in [10]. A record of the constraints which have not been revised is kept to ensure that their domains are arc-consistent. When the membership degree of a variable changes, all the constraints related to that variable are added to the *CheckList* so that they can be revised for arc-consistency. The variable *ConsSup* saves the maximum consistency value for a solution of the FCSP $\mathcal{P}$. When applying RevisedFuzzyConstraint method on a constraint C_k the constraint is propagated through the domains of the variables in S_k by modifying their membership degrees. For every value $v \in D_i$ of a variable $x_i \in S_k$, the membership degree $\mu_{x_i}(v)$ is replaced by the supremum in the right hand side of Eq. (5). If $\mu_{x_i}(v)$ is equal to zero then v is removed from the domain of D_i. If μ_{x_i} changes, then the variable x_i is marked as changed in order to check the arc-consistency with respect to other constraints.

For each variable x_i the initial membership function μ_{x_i} is a constant function over D_i equal to one. When more knowledge about the degree of satisfaction of the relations involving x_i is acquired, the membership μ_{x_i} is modified to incorporate this new knowledge.

5. Extension of the FCSP model to handle complex concept nodes

To deal with complex concept nodes representing groups of regions satisfying some relations between them, Algorithm 1 has to be extended and adapted.

Algorithm 1: FAC-3 algorithm used for determining the arc-consistency of a FCSP [17].

Input: A FCSP $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$
Output: Computes the arc-consistent closure of $\mathcal{P}$ if it exists, otherwise returns Failure

ConsSup $= 1$
ToCheck $\longleftarrow \mathcal{C}$
while *ToCheck* $\neq \emptyset$ **do**
    ToCheck $\longleftarrow$ *ToCheck* $\setminus \{C_k\}$ // Select C_k from *ToCheck*
    foreach $x_{k_i} \in S_k$ **do**
        Changed$[k_i] \longleftarrow$ *false*
    result $\longleftarrow$ `ReviseFuzzyConstraint`(C_k) // see Algorithm 2
    if *result* $=$ *EmptyDomain* **then**
        return Failure ;
    if *result* $=$ *Changed* **then**
        foreach $C_l \neq C_k$ *such that there is* $x_j \in S_k \cap S_l$ *and Changed*$[j] =$ *true* **do**
            ToCheck $\longleftarrow$ *ToCheck* $\cup \{C_l\}$
return *ConsSup* ;

Algorithm 2: *ReviseFuzzyConstraint* method.

Input: C_k, *Changed*, *ConsSup*
Output: Propagates C_k and marks variables whose domain has changed in the *Changed* vector

foreach *variable* $x_i \in S_k$ **do**
    remove from R_k every tuple A such that $A\downarrow_i \notin D_i$
$Height \longleftarrow 0$
$result \longleftarrow NoChange$
foreach *variable* $x_i \in S_k$ **do**
    foreach $v \in D_i$ **do**
        $newDegree \longleftarrow 0$
        foreach *tuple* $A = (a_1, \dots, a_m)$ *in the domain of* R_k *such that* $v = A\downarrow_i$ **do**
            $eval \longleftarrow \min(\mu_{R_k}(a_1, \dots, a_m), \min_{\substack{j\in\{1,\dots,m\}\\ j\neq i}} \mu_{x_j}(a_j))$
            $height \longleftarrow \max(eval, height)$
            $newDegree \longleftarrow \max(eval, newDegree)$
    if $newDegree = 0$ **then**
        Delete v from D_i
        if $D_i = \emptyset$ **then** return *EmptyDomain*
    if $newDegree < \mu_{x_i}(v)$ **then**
        $Changed[i] \longleftarrow true$
        $\mu_{x_i}(v) \longleftarrow newDegree$
        $result \longleftarrow Changed$
$ConsSup \longleftarrow \min(ConsSup, Height)$
return *result*

A complex concept node can be considered dually either as a constraint or as a variable. It is seen as a variable when it is viewed as an object (a group is then considered as one complex object) that can satisfy spatial relations with other objects or groups. It is seen as a constraint when we evaluate the relations or spatial properties that should be satisfied within the group of objects. Thus, we propose to define this constraint/variable by considering its two parts: (1) as a relation representing the conjunction of all the conditions that should be satisfied inside the complex concept node, and (2) as a membership degree in the domain of groups which depends on the satisfaction of the conditions inside the nested node, as well as the satisfaction of the relations of the group with other objects. We differentiate between the complex concept nodes which represent a group endowed with a spatial property, as alignment, and those which are not.

5.1. Dealing with complex nodes endowed with a spatial property

In our vocabulary we consider only the alignment relation as a spatial property. Hence in this section we only detail this relation, but the proposed formalism can be extended to other spatial properties. For the sake of simplicity, the aligned groups of objects are only considered for objects belonging to the same concept type. Let x_i be the variable representing the type of the objects involved in the alignment, and let D_i be its domain. Let x_g denote the variable which represents the aligned group of objects considered as a variable, and let C_w represent the constraint of alignment of the group.

When the group is seen as a variable x_g, the domain D_g is composed of the groups of objects of type x_i which are considered to be aligned. These groups can be extracted using the algorithm proposed in [46]. D_g is a subset of the power set of D_i. For a group $V = \{v_1, \dots, v_p\} \in D_g$ the membership degree $\mu_{x_g}(V)$ depends on three factors:

1. the degree of alignment of V [46],
2. the degree of satisfaction of the spatial constraints (spatial relations) that are supposed to be satisfied by x_g, and
3. the membership degree of its members to D_i.

(a) Conceptual graph. (b) Domains.

Fig. 4. Illustration of the dual characteristic of the group V of aligned objects. Let $D_1 = \{v_1, v_2, v_3\}$, $D_2 = \{b_1\}$. Suppose that for every $v_j \in V$ we have $\mu_{x_1}(v_j) = 1.0$. Before we have any information about the satisfaction of the parallel relation we have $\mu_{x_g}(V) = \mu_{R_{\text{aligned}}}(V) = 1$, where x_g is the variable that represents the group, and $C_{\text{aligned}} = \langle R_{\text{aligned}}, S_{\text{aligned}}\rangle$ is the alignment constraint of the group. Suppose that we find that $\mu_{R_{\text{parallel}}}(V, b) = 0.3$, then the membership degree of V belonging to the domain of x_g becomes $\mu_{x_g}(V) = 0.3$, while we still have $\mu_{R_{\text{aligned}}}(V) = 1$.

When the groups are extracted we do not have any information about the satisfaction of the spatial relations which involve V. Therefore, its initial degree of satisfaction is equal to the conjunction of the factors 1 and 3, listed above, using the minimum as conjunction:

$$\mu_{x_g}(V) = \min\Big(\mu_{\text{alig}}(V), \min_{j\in\{1,\dots,p\}} \mu_{x_i}(v_j)\Big) \tag{6}$$

where $\mu_{\text{alig}}(V)$ is the degree of alignment of the objects in the group V [46]. The membership degree $\mu_{x_g}(V)$ changes as more information about the factors 2 and 3 is acquired.

When the group is considered as a constraint C_w, it evaluates the property of alignment of a group. As before, a constraint C_k is defined by a pair $\langle R_k, S_k\rangle$. Usually a relation representing a constraint is defined as a subset of a Cartesian product of the domains of its variables, and the number of domains and variables is fixed. However, this is not the case for the constraint C_w, since each group can have a different number of elements, which is a priori unknown. Therefore, to properly define the relation representing this constraint, it is necessary to specify for each $p \geq 3$ a relation with arity p, to define the groups of aligned objects in D_i^p. Due to the lack of knowledge about the number of elements in a group, we use as a simplification the same notation to define the relations for each possible arity. Then the degree of satisfaction of the relation of a tuple $(v_1, \dots, v_p) \in D_i^p$ is equal to the degree of alignment of the set $V = \{v_1, \dots, v_p\}$, and of the conjunction of the degrees μ_{x_i} of its elements:

$$\mu_{R_w}(v_1, \dots, v_p) = \min\Big(\mu_{\text{alig}}(V), \min_{j\in\{1,\dots,p\}} \mu_{x_i}(v_j)\Big) \tag{7}$$

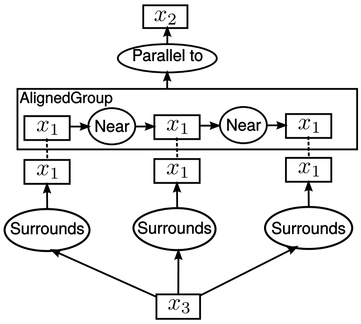
and $S_w = \{x_i\}$, where x_i represents the variable of the objects that satisfy the relation (recall that the relation is only defined for one type of object). At the beginning both μ_{R_w} and μ_{x_g} are identical. However, as we make D_g arc-consistent with respect to other constraints, the values of μ_{R_w} and μ_{x_g} start to differ. Fig. 4 shows an example where $\mu_{R_w} = \mu_{x_g}$ when we do not have any information about the parallel relation. Once we have calculated the degree of parallelism between V and b_1, the value of μ_{x_g} changes, while μ_{R_w} remains the same.

For a complex concept node representing an aligned group of objects, it is necessary to consider both degrees $\mu_{R_w}(V)$ and $\mu_{x_g}(V)$ because they represent different types of information. The membership degree $\mu_{R_w}(V)$ remains independent of the relations that the group satisfies, while the membership degree $\mu_{x_g}(V)$ varies according to the interaction of the group with other elements.

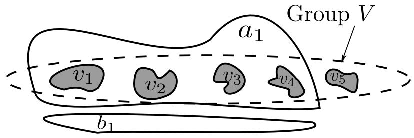
The dual characteristic of this constraint requires a careful evaluation of its satisfaction. In addition the constraint has to be evaluated before it is evaluated as a variable. In the following we present several considerations that should be taken into account when making D_g arc-consistent.

5.2. Adapting arc-consistency for groups of aligned objects

We first discuss the considerations dealing with the changes in the membership function μ_{x_i}, and then, those referring to the situation when the group is seen as an object.



(a) Conceptual graph.



(b) Domains.

Fig. 5. The element v_5 belonging to the group $V = \{v_1, v_2, v_3, v_4, v_5\}$ does not satisfy the relation "topologically surrounds" with a_1, although this relation should be satisfied by all the members of the group. However, there are subgroups of V which satisfy all the constraints, such as $\{v_1, v_2, v_3, v_4\}$.

A member $v_j \in V$ does no longer belong to D_i Let us assume that there is $v_j \in V$ for which $\mu_{x_i}(v_j) = 0$. By Eq. (6) this implies that $\mu_{x_g}(V) = 0$. However, it is possible that there is an aligned subgroup $U \subseteq V \setminus \{v_j\}$ which is part of a consistent solution. For example, if we want to find the instantiations of the conceptual graph of Fig. 5(a) in the image of Fig. 5(b), the corresponding FCSP $\mathcal{P}$ is formulated as:

- $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$, where x_1, x_2 and x_3 correspond to the concept nodes of Fig. 5(a) and x_4 corresponds to the alignment variable.
- $\mathcal{D} = \{D_1, D_2, D_3, D_4\}$, where $D_1 = \{v_1, v_2, v_3, v_4, v_5\}$, $D_2 = \{b_1\}$, $D_3 = \{a_1\}$ and $D_4 = \{V\}$ for $V = \{v_1, v_2, v_3, v_4, v_5\}$.
- $\mathcal{C} = \{C_{\text{surround}}, C_{\text{parallel\_to}}, C_{\text{aligned}}\}$ where $C_{\text{surround}} = \langle \mu_{\text{surround}}, \{x_3, x_1\}\rangle$, $C_{\text{parallel\_to}} = \langle \mu_{||}, \{x_4, x_2\}\rangle$ and $C_{\text{aligned}} = \langle \mu_{\text{alig}}, \{x_1\}\rangle$. Note that we do not consider the constraint referring to the "near" relation inside the aligned group, because it is embedded in the "aligned" relation (see Section 3.3).

Let us try to make $\mathcal{P}$ arc-consistent by following the procedure presented in the previous section. When we propagate the constraint C_{surround} through its domain, we obtain $\mu_{x_1}(v_5) = 0$ (because v_5 is not surrounded by a_1), and therefore the fuzzy set μ_{x_1} over D_1 is modified. Thus, the constraint C_{aligned} has to be checked, because $x_1 \in S_{\text{aligned}}$. When we revise C_{aligned} the membership function μ_{x_4} is updated, and $\mu_{x_4}(V) = 0$. Hence, D_4 is empty and no solution is found. Nevertheless, the group $U = \{v_1, v_2, v_3, v_4\} \subseteq V$ is part of a consistent instantiation. Thus, eliminating V without considering if any of its subgroups satisfied the relations was a too early decision. To avoid this, we propose to use a greedy strategy, where we start with a group of N elements and reconsider the group $V \setminus U$ when the elements of $U \subset V$ does no longer belong to D_i. This strategy is more efficient than considering all the possible subgroups of V for the domain of D_g.[2] One should notice that the proposed strategy does not evaluate all the possible groups but only some of them, therefore we would not obtain all the possible solutions. In our application we are interested in the largest group which satisfies the constraint, and therefore once we find a solution it is not necessary to consider its subgroups. However, if one is interested in all the possible solutions then it is necessary to evaluate all the possible groups.

The membership function μ_{x_i} has changed Let $v_j \in V$ be an element of a group. Suppose that after checking the arc-consistency condition the membership degree of v_j to D_i has decreased. The decrease of $\mu_{x_i}(v_j)$ implies that it is less possible that x_i takes the value v_j than initially, i.e. at the time the aligned group was extracted. To integrate this new information into the membership degree of V, i.e. $\mu_{x_g}(V)$, we propose to add $V \setminus \{v_j\}$ to D_g (without removing V), and update μ_{x_g}. When adding this new group to D_g, the relations that involve x_g have to be reevaluated, since D_g has changed.

[2] The greedy strategy has a worst case complexity of evaluating $N - 3$ groups, while for the other strategy in the worst case scenario, the size of the domain of the constraint is $\sum_{k=3}^{N} \binom{N}{k}$.

The group V, considered as a variable, does not satisfy a constraint Due to the fact that we are considering only some of the groups, we need to consider what happens to the aligned subgroups of a group V when V does not satisfy a constraint that should be satisfied by the members of D_g.

Let us first remember that when a spatial relation is represented as a fuzzy landscape, the degree of satisfaction of a target object is measured using the mean (Eq. (1)). Now, let us consider the situation where x_g is the target object of the relation.

Proposition 1. *Let $V = \{v_1, \dots, v_p\} \in D_g$ be an aligned group and $C_t = \langle \mu_{R_t}, S_t \rangle$ be a binary constraint, such that $S_t = \{x_g, x_w\}$, and R_t is a relation from T_R. Let $b \in D_w$, suppose that x_g is the target object of the relation represented by C_t and that $\mu_{R_t}(b, V) = 0$. Then for every aligned subgroup $U \subset V$ we have $\mu_{R_t}(b, U) = 0$.*

Proof. We consider each of the possible binary relations in T_R:

(a) Let R_t be a relation which produces a fuzzy landscape denoted by γ_R.
If $\mu_{R_t}(b, V) = 0$, then it means that for every pixel q belonging to V we have $\gamma_R(q) = 0$. This holds in particular for the pixels q in the region $U \subset V$, therefore $\mu_{R_t}(b, U) = 0$.
(b) If R_t corresponds to the adjacency relation [4], and $\mu_{\text{adjacency}}(b, V) = 0$, then the dilation of b does not intersect V, and it does not intersect U either, and $\mu_{\text{adjacency}}(b, U) = 0$.
(c) If R_t corresponds to the parallel relation [46], and $\mu_{\text{parallel}}(b, V) = 0$, then it is either because the fuzzy landscape component is zero or because the orientations are not similar. If the fuzzy component is zero, then this is also true for U for the same reason as in (a). If the orientations are not similar then the orientation of U is not similar to the one of b, since the orientation of U can be considered equal to the orientation of V, and therefore $\mu_{\text{parallel}}(b, U) = 0$. □

Now, let us consider the situation where x_g is the reference object of the relation.

Proposition 2. *Let $V = \{v_1, \dots, v_p\} \in D_g$ be an aligned group and $C_t = \langle \mu_{R_t}, S_t \rangle$ be a binary constraint, such that $S_t = \{x_g, x_w\}$, and R_t is a relation from T_R. Let $b \in D_w$, and suppose that x_g is the reference object of the relation represented by C_t and that $\mu_{R_t}(V, b) = 0$. Then for every aligned subgroup $U \subset V$ we have $\mu_{R_t}(U, b) = 0$.*

Proof. We consider each of the possible binary relations in T_R:

(a) If R_t is a metric relation which produces a fuzzy landscape, then R_t produces a fuzzy landscape which is increasing with respect to the reference object. Therefore, the landscape produced by U is contained in the one produced by V. So, $\mu_{R_t}(U, b) = 0$.
(b) If R_t is the adjacency relation, then the dilation of V does not intersect b. Since the dilation is increasing the dilation of U is a subset of the dilation of V, as U is a subset of V, therefore the dilation of U does not intersect b and $\mu_{\text{adjacency}}(U, b) = 0$
(c) If R_t is the parallel relation, and $\mu_{\text{parallel}}(V, b) = 0$, then it means that either the part of the relation that is modeled as a fuzzy landscape is not satisfied or the orientation condition is not satisfied. In the case when the first condition is not satisfied, then as in the previous case it is not satisfied by U. If the orientation condition is not satisfied then it is not satisfied by U either. □

In [5] several definitions for the relation "between" are proposed. In our work we use the relation based on directional dilation. All the models proposed in [5] produce a fuzzy landscape, so the following propositions are valid for all these models. Moreover, the fuzzy landscape to represent the relation "between" a group of objects $V = \{v_1, \dots, v_p\}$ and another object b is modeled as the union of the fuzzy landscapes created for each element of the group. Hence, using similar arguments we can extend the previous propositions to deal with the ternary relation "between".

Proposition 3. *Let $V = \{v_1, \dots, v_p\} \in D_g$ be an aligned group and C_t be a constraint representing the relation "between", such that $S_t = \{x_g, x_w, x_z\}$. Let $a \in D_w$, $b \in D_z$, and suppose that V is one of the reference objects of the relation and that $\mu_{R_t}(V, a, b) = 0$. Then for every aligned subgroup $U \subset V$ we have $\mu_{R_t}(U, a, b) = 0$.*

Proposition 4. *Let $V = \{v_1, \ldots, v_p\} \in D_g$ be an aligned group and C_t be a constraint representing the relation "between", such that $S_t = \{x_g, x_w, x_z\}$. Let $a \in D_w$, $b \in D_z$, and suppose that V is the target object of the relation and that $\mu_{R_t}(a, b, V) = 0$. Then for every aligned subgroup $U \subset V$ we have $\mu_{R_t}(a, b, U) = 0$.*

After considering all the possible situations where a group does not satisfy a constraint, we can conclude that when a group V does not satisfy a constraint, then none of its aligned subgroups satisfies it. Therefore, we can remove V from D_g without further considering its subgroups.

5.3. Dealing with groups not endowed with a spatial property

The other types of complex concept nodes are the groups which have a conceptual graph not representing an alignment (or more generally any other relation within the group). Suppose that we want to represent a complex concept node with child concept nodes represented by the set of variables $\{x_1, \ldots, x_l\}$ and child relations $\{R_1, \ldots, R_t\}$.

When the group is considered as a variable x_g its domain is $D_g = D_1 \times \ldots \times D_l$. Let $V = (v_1, \ldots, v_l) \in D_1 \times \ldots \times D_l$. Its membership degree $\mu_{x_g}(V)$ depends on:

- the degree of satisfaction of the child relations,
- the degree of satisfaction of the spatial constraints (spatial relations) that are supposed to be satisfied by x_g, and
- the membership degree of its members to their respective domains.

As in Section 5.1, initially we do not have any information about the degree of satisfaction of V with other regions. Thus its initial degree $\mu_{x_g}(V)$ is equal to:

$$\mu_{x_g}(V) = \min\left[\bigwedge_{j=1}^{t} \mu_{R_j}(V \downarrow_{S_j}), \min_{h \in \{1,\ldots,l\}} \mu_{x_h}(v_h)\right] \tag{8}$$

where S_j is the set of variables involved in R_j, and $\bigwedge$ is a t-norm (the minimum has been used in our experiments).

When the group is considered as a constraint C_w, it evaluates that all the conditions inside the complex concept node are satisfied. Unlike the alignment case, we know in advance which are the members inside the complex concept node. Therefore the set of variables in C_w is $S_w = \{x_1, \ldots, x_l\}$, which is the union of the sets of variables of its child relations: $S_w = \bigcup_{j=1}^{t} S_j$. For a tuple $V = (v_{w_1}, \ldots, v_{w_l}) \in D_{w_1} \times \ldots \times D_{w_l}$ the degree of satisfaction of the relation is:

$$\mu_{R_w}(v_{w_1}, \ldots, v_{w_l}) = \min\left[\bigwedge_{j=1}^{t} \mu_{R_j}(V \downarrow_{S_j}), \min_{h \in \{w_1,\ldots,w_l\}} \mu_{x_h}(v_h)\right]. \tag{9}$$

As for the aligned groups, the degrees $\mu_{R_w}(V)$ and $\mu_{x_g}(V)$ are equal at the beginning but as more information is acquired, the degrees $\mu_{R_w}(V)$ and $\mu_{x_g}(V)$ may become different.

Behavior of the members of a group, when the group is the target object of a metric relation Let $V = \{v_1, \ldots, v_l\} \in D_g$ be a group (composed of objects that may be aligned or not) and C_t be a constraint representing a metric relation R_t between the group and another object or group, such that $S_t = \{x_g, x_u\}$. Suppose that R_t is modeled as a fuzzy landscape. Let $b \in D_u$. The degree of satisfaction of the relation is non-zero if and only if there exists $v_i \in V$ such that $\mu_{R_t}(v_i, b) > 0$. Moreover, if the elements of the group have a high satisfaction degree of R_t, then the group has a high degree of satisfaction of R_t.

Thus if the model conceptual graph has a relation node representing a metric relation R_t where the target object is a complex concept node representing a group and the reference object is another concept node c_{ref} (which can be complex or not), and R_t is modeled as a fuzzy landscape, then we can add a relation node having the same type as R_t between each concept node inside the complex concept node and the node c_{ref} as reference (as shown in Fig. 6). This is a heuristic technique that we propose to help us to find an instantiation of the group with a high satisfaction degree of the relation R_t.

Other relations such as adjacency have a different behavior, and for a group to be adjacent to an object it is sufficient that one of its members is adjacent to the object (usually not all objects in the group satisfy the relation).

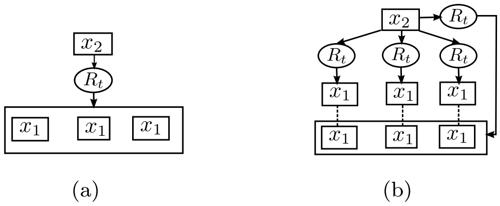


Fig. 6. Illustration of the behavior of a group when the group is the target of a metric relation modeled as a fuzzy landscape. (a) Initial conceptual graph. (b) Modified graph after adding relation nodes having the relation R_t between x_1 and each member of the group.

Algorithm 3: Basic algorithm used for arc-consistency checking in a nested constraint network with complex concept nodes.

Input: A constraint network $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$
Output: Computes the arc-consistent closure of $\mathcal{P}$ if it exists, otherwise returns Failure

1 *ConsSup* = 1
2 *ToCheck* = $\mathcal{C}$ `// Initialize the` *Instantiated* `vector, mark as true all the variables which do not represent a group`
3 **foreach** *variable* $x_i \in \mathcal{X}$ **do**
4   **if** x_i *does not represent a group* **then** *Instantiated*$[i]$ = *true* **else** *Instantiated*$[i]$ = *false*
5 **foreach** $C_k \in \mathcal{C}$ **do**
6   *FirstEvaluation*$[k]$ = *false*
7 **while** *ToCheck* $\neq \emptyset$ **do**
8   *ToCheck* $\longleftarrow$ *ToCheck* $\setminus \{C_k\}$ `// Select` C_k `from` *ToCheck*
9   **foreach** *variable* $x_{k_i} \in S_k$ **do**
10   *Changed*$[k_i]$ = *false*
11   **if** C_k *represents a group* **then**
12   *result* = `ReviseGroupConstraint`(C_k) `// see Algorithm 5`
13   **else**
14   **if** C_k *represents an alignment* **then**
15   *result* = `ReviseAlignmentConstraint`(C_k) `// see Algorithm 7`
16   **else**
17   *result* = `ReviseSimpleConstraint`(C_k) `// see Algorithm 4`
18
19   **if** *result* = *EmptyDomain* **then**
20   return Failure
21   **if** *result* = *Changed* **then**
22   **foreach** $C_l \neq C_k$ *such that there is* $x_i \in S_l$ *with Changed*$[i]$ = *true* **do**
23   *ToCheck* = *ToCheck* $\cup \{C_l\}$
24   *Change*$[i] \longleftarrow$ *false*
25   **if** C_k *is inside a nested constraint* C_l *or is related to a variable that represents a nested node represented by the constraint* C_l **then**
26   *ToCheck* = *ToCheck* $\cup \{C_l\}$
27
28 return *ConsSup* ;

5.4. Proposed algorithm

In this section, we propose a new algorithm based on the FAC-3 algorithm which deals with nested constraints. The set of constraints $\mathcal{C}$ contains the constraints representing the relation nodes and the complex concept nodes.

The main algorithm presented in Algorithm 3 has the same structure as the one used in the original FAC-3 (see Algorithm 1). The main difference with respect to the FAC-3 algorithm is the adaptation of the revise function to each type of constraint. We use the methods ReviseGroupConstraint (Algorithm 5) and ReviseAlignmentConstraint (Algorithm 7) for the constraints representing complex concept nodes. The method ReviseSimpleConstraint (Algorithm 4) is used for constraints representing relation nodes. We refer to these three methods as Revise methods. When updating the *ToCheck* list (lines 21 to 26 of Algorithm 3) we take into account the dual interpretation of the complex concept

Algorithm 4: *ReviseSimpleConstraint*: revise algorithm for those constraints which do not represent alignments nor groups, sub-part of Algorithm 3.

Input: a constraint C_k, *Changed*, *ConsSup*
Output: Propagates C_k if possible and marks variables whose domain has changed; returns EmptyDomain if a domain becomes empty, NoChange if no domain has been modified, otherwise Changed

// Verify that all the variables in S_k are instantiated
if *Exists* $x_i \in S_k$ *such that Instantiated*$[i] =$ *false* **then**
    return *NoChange*
if *FirstEvaluation*$[k]$ **then** // Create the domain for R_k
    Let $S_k = \{x_1, \dots, x_m\}$
    foreach $A = (a_1, \dots, a_m) \in D_1 \times \dots \times D_m$ **do**
        if $\mu_{R_k}(a_1, \dots, a_m) > 0$ **then** Add A to the domain of R_k
    FirstEvaluation$[k] \longleftarrow$ *false*
else
    foreach *variable* $x_i \in S_k$ **do**
        remove from the domain of R_k every tuple $A = (a_1, \dots, a_m)$ such that $a_i \notin D_i$
// Make the domains of the variables in S_k arc-consistent with respect to C_k
result $\longleftarrow$ *NoChange*
height $\longleftarrow 0$
foreach *variable* $x_i \in S_k$ **do**
    foreach $v \in D_i$ **do**
        NewDegree $= 0$
        foreach $A = (a_1, \dots, a_m)$ *in the domain of* R_k *such that* $a_i = v$ **do**
            eval $\longleftarrow \min(\mu_{R_k}(a_1, \dots, a_m), \min_{s \in \{1,\dots,m\}, s \neq i} \mu_{x_s}(a_s))$
            height $\longleftarrow \max($*eval*, *height*$)$
            NewDegree $\longleftarrow \max($*NewDegree*, *eval*$)$
        if *NewDegree* $= 0$ **then**
            Delete v from D_i
            if $D_i = \emptyset$ **then** return *EmptyDomain*
        if *NewDegree* $< \mu_{x_i}(v)$ **then**
            $\mu_{x_i}(v) \longleftarrow$ *NewDegree*
            result $\longleftarrow$ *Changed*
            Changed$[i] \longleftarrow$ *true*
ConsSup $\longleftarrow \min($*ConsSup*, *height*$)$
return *result*

nodes, either as a variable or as a constraint. If the domain of a variable which is also a constraint has changed, then the corresponding constraint should be added to the *ToCheck* list.

The vector *Instantiated* indicates whether the domain of the variable has been already instantiated or not. This vector is needed to verify that the variables which represent a complex concept have been instantiated when the respective constraint is evaluated. In the Revise methods, we first verify that all the variables involved in the relation have been instantiated. When it is not the case, the constraint is not propagated.

ReviseSimpleConstraint The method ReviseSimpleConstraint in Algorithm 4 is composed of two parts. In the first part, if the constraint has not been evaluated, then it is evaluated for the first time and the domain of the relation is created. To create this domain we evaluate the relation according to the way it is modeled. If it is modeled as a fuzzy landscape, then we compute the landscape for each of the elements in the domain of the reference object, and evaluate the relation with all the elements in the domain of the target objects. Otherwise, if the relation is modeled as a number (satisfaction degree), we find the satisfaction degree for each possible tuple in the relation domain. To update we use the same strategy as in the Revise method of the FAC-3 algorithm (see Algorithm 2).

ReviseGroupConstraint The method ReviseGroupConstraint in Algorithm 5 is also composed of two parts. In the first part, we create or update the domain of the relation (Algorithm 6). This domain is created by making an exhaustive

Algorithm 5: *ReviseGroupConstraint*: revise algorithm for those constraints which represent a group, sub-part of Algorithm 3.

```
Input: a constraint C_k, Changed, ConsSup
Output: Propagates C_k and marks variables whose domain has changed; returns EmptyDomain if a domain becomes empty, NoChange if
        no domain has been modified, otherwise Changed
Let x_g be the variable representing the nested group and D_g its domain
if Exists x_i ∈ S_k such that Instantiated[i] = false then
    return NoChange
height ⟵ 0
result ⟵ CreateOrUpdateGroupDomain(C_k, D_g)                    // See Algorithm 6
if D_g = ∅ then
    return EmptyDomain
// Make the domains of the variables in S_k arc-consistent with respect to C_k
foreach variable x_i ∈ S_k do
    foreach v ∈ D_i do
        NewDegree = 0
        foreach A = (a_1, ..., a_l) in the domain of R_k such that a_i = v do
            height ⟵ max(μ_{x_g}(A), height)
            NewDegree ⟵ max(NewDegree, μ_{x_g}(A))
        if NewDegree = 0 then
            Delete v from D_i
            if D_i = ∅ then return EmptyDomain
        if NewDegree < μ_{x_i}(v) then        // Update Changed and the membership value of v
            μ_{x_i}(v) ⟵ NewDegree
            result ⟵ Changed
            Changed[i] ⟵ true
            Changed[g] ⟵ true

ConsSup ⟵ min(ConsSup, height)
return result
```

search of its domain and only adding the tuples for which the satisfaction degrees of all the child relations of the nested node are non-zero. The updating of the domain follows the same strategy as the original FAC-3 algorithm. The second part of Algorithm 5 corresponds to the updating of the domain of the variables representing the child nodes. We update the value of the membership function of x_i by considering the value of the corresponding constraint. The membership function μ_{x_g} considers the interactions between the members of the group as well as the satisfaction of the relations between the group and other variables, which may impact the satisfaction of the membership function of the child nodes.

ReviseAlignmentConstraint In the first part of the ReviseAlignmentConstraint method in Algorithm 7, we construct or update the domain of the constraint and its respective variables by calling the method CreateOrUpdateAlignmentDomain (Algorithm 8). To create the domain we search for the aligned groups of objects in D_k we apply the algorithm proposed in [46]. When we update the domain D_g we check for each element a_j of each group A its membership degree $\mu_{x_i}(a_j)$. If it is equal to zero then the aligned subgroups in $B_t \subseteq A \setminus \{a_j\}$ are added to the domain D_g. As we mentioned in Section 5.2 we do not make an exhaustive search of the aligned objects but we only look for the longest aligned groups that satisfy the constraints in our model. This is a heuristic technique to evaluate less constraints dealing with aligned groups of objects.

The second part of ReviseAlignmentConstraint propagates C_k in the domain D_i. For each value $a_j \in D_i$ we verify that there exists a group in D_g which contains it. To do this we replace the membership degree of $\mu_{x_i}(a_j)$ by:

$$\mu_{x_i}(a_j) = \max_{\{V \in D_g | a_j \in V\}} \mu_{x_g}(V)$$

Algorithm 6: *CreateOrUpdateGroupDomain*: sub-part of Algorithm 5.

Input: A constraint C_k representing a nested group, D_g domain of the corresponding variable
Output: Creates or updates the domain of C_k and of x_g

Let $S_k = \{x_1, \ldots, x_l\}$ be the child variables, $T_k = \{C_{t_1}, \ldots, C_{t_p}\}$ be the constraints representing the child relations.
if *FirstEvaluation*[k] **then** // Create the domain for R_k and x_g
  foreach $A = (a_1, \ldots, a_l) \in D_1 \times \ldots \times D_l$ **do**
    $\mu_{R_k}(a_1, \ldots, a_l) \longleftarrow \bigwedge_{j=t_1}^{t_p} \mu_{R_j}(a_1, \ldots, a_l)$
    if $\mu_{R_k}(a_1, \ldots, a_l) > 0$ **then**
      Add A to the domain of R_k and to D_g
      $\mu_{x_g}(A) \longleftarrow \min[\min_{s=1,\ldots,l} \mu_{x_s}(a_s), \mu_{R_k}(a_1, \ldots, a_l)]$
  FirstEvaluation[k] $\longleftarrow$ *false*, *Instantiated*[g] $\longleftarrow$ *true*, *Changed*[g] $\longleftarrow$ *true*, *result* $\longleftarrow$ *Changed*
else
  foreach *variable* $x_i \in S_k$ **do**
    remove from the domain of R_k every tuple $A = (a_1, \ldots, a_l)$ such $a_i \notin D_i$
  foreach $A = (a_1, \ldots, a_l) \in D_g$ **do**
    //Update the membership degree
    NewDegree $\longleftarrow \min[\min_{s \in \{1,\ldots,l\}} \mu_{x_s}(a_s), \mu_{R_k}(a_1, \ldots, a_l), \mu_{x_g}(A)]$
    if *NewDegree* $= 0$ **then**
      Delete A from R_k and from D_g
    if *NewDegree* $< \mu_{x_g}(A)$ **then** // Update *Changed* and the membership degree of A
      $\mu_{x_g}(A) \longleftarrow$ *NewDegree*
      Changed[g] $\longleftarrow$ *true*
      result $\longleftarrow$ *Changed*
return *result*

Algorithm 7: *ReviseAlignmentConstraint*: revise algorithm for those constraints which represent an alignment, sub-part of Algorithm 3.

Input: a constraint C_k, *Changed*, *ConsSup*
Output: Propagates C_k and marks variables whose domain has changed; returns EmptyDomain if a domain becomes empty, NoChange if no domain has been modified, otherwise Changed

Let x_g be the variable representing the nested group and D_g its domain
if *Exists* $x_i \in S_k$ *such that Instantiated*[i] $=$ *false* **then**
  return *NoChange*
height $\longleftarrow 0$
result $\longleftarrow$ `CreateOrUpdateAlignmentDomain`(C_k, D_g) // See Algorithm 8
if $D_g = \emptyset$ **then**
  return *EmptyDomain*
// Make the domains of the variables in S_k arc-consistent with respect to C_k
foreach $v \in D_i$ **do**
  NewDegree $= 0$
  foreach $A \in D_g$ *such that* $v \in A$ **do**
    height $\longleftarrow \max(\mu_{x_g}(A), \textit{height})$
    NewDegree $\longleftarrow \max(\textit{NewDegree}, \mu_{x_g}(A))$
  if *NewDegree* $= 0$ **then**
    Delete v from D_i
    if $D_i = \emptyset$ **then** return *EmptyDomain*
  if *NewDegree* $< \mu_{x_i}(v)$ **then** // Update *Changed* and the membership degree of v
    $\mu_{x_i}(v) \longleftarrow$ *NewDegree*
    result $\longleftarrow$ *Changed*
    Changed[i] $\longleftarrow$ *true*
    Changed[g] $\longleftarrow$ *true*
ConsSup $\longleftarrow \min(\textit{ConsSup}, \textit{height})$
return *result*

Algorithm 8: *CreateOrUpdateAlignmentDomain*: sub-part of Algorithm 5.

Input: A constraint C_k representing an aligned group, D_g domain of the corresponding variable, and D_i the domain of the objects involved in the alignment.

Output: Creates or updates the domain of C_k and of x_g; returns EmptyDomain if a domain becomes empty, NoChange if no domain has been modified, otherwise Changed

Let $S_k = \{x_i\}$ be the variable involved in the alignment.
domainStatus $\longleftarrow$ *NoChange*
`// Create the domain for` R_k `and` x_g
if *FirstEvaluation*$[k]$ **then**
  Find $\mathfrak{G}$, the group of aligned objects belonging to the domain D_i
  foreach $A = \{a_1, \dots, a_p\} \in \mathfrak{G}$ **do**
    Let $\mu_{\text{alig}}(A)$ be the degree of alignment of A
    $\mu_{R_k}(A) \longleftarrow \min[\min_{s\in\{1,\dots,p\}} \mu_{x_i}(a_s), \mu_{\text{alig}}(A)]$
    Add A to D_g and add the tuple $(a_1, \dots, a_p)$ to the domain of R_k
    $\mu_{x_g}(A) \longleftarrow \min[\min_{s\in\{1,\dots,p\}} \mu_{x_i}(a_s), \mu_{\text{alig}}(A)]$
  FirstEvaluation$[k] \longleftarrow$ *false*; *Instantiated*$[g] \longleftarrow$ *true*
  if $D_k \neq \emptyset$ **then**
    Changed$[g] \longleftarrow$ *true*; *domainStatus* $\longleftarrow$ *Changed*
  else return *EmptyDomain* return *domainStatus*
`// Revise the alignment condition on each group`
else
  foreach $A = \{a_1, \dots, a_p\} \in D_g$ **do**
    if *There is an element* $a_j \in A$ *which does no longer belong to* D_i **then**
      Remove A from D_g and R_k
      Let $\mathfrak{G}_A$, be the group of aligned objects belonging to $A \setminus \{a_j\}$
      foreach $B = \{b_1, \dots, b_m\} \in \mathfrak{G}_A$ **do**
        $\mu_{R_k}(b_1, \dots, b_m) \longleftarrow \min[\min_{s\in\{1,\dots,p\}} \mu_{x_i}(b_s), \mu_{\text{alig}}(B)]$
        Add the set B to D_g
        Add the tuple $(b_1, \dots, b_m)$ to the domain of R_k
        $\mu_{x_g}(B) \longleftarrow \min[\min_{s\in\{1,\dots,m\}} \mu_{x_i}(b_s), \mu_{\text{alig}}(B)]$
        Changed$[g] \longleftarrow$ *true*
        domainStatus $\longleftarrow$ *Changed*
        Set to *true* the value of *FirstEvaluation* for all the constraints involving x_g

5.5. Illustration of the algorithm

We illustrate how the algorithm works on the following example. We assume that the image in Fig. 7(a) has already been segmented and the regions have been labeled as shown in Fig. 7(b). The aim is to find the instantiations in the image of the graph in Fig. 7(d), which is built over the vocabulary shown in Fig. 7(c). For the sake of simplicity, we only show one coreference link of the houses in the aligned group, but all the houses have the same coreference link as the one illustrated in the figure.

The corresponding CSP problem is formulated as:

- $\mathcal{X} = \{x_{\text{shadow}}, x_{\text{house}}, x_{\text{road}}, x_{\text{pool}}, x_{\text{garden}}, x_{\text{group\_houses}}\}$ containing the variables representing the concept nodes of the conceptual graph, where $x_{\text{group\_houses}}$ is the variable representing the group of aligned houses.
- $\mathcal{C} = \{C_{\text{direction}}, C_{\text{adjacent\_1}}, C_{\text{between}}, C_{\text{near}}, C_{\text{surrounds}}, C_{\text{adjacent\_2}}, C_{\text{aligned}}\}$, where $C_{\text{direction}} = \langle \mu^{135^0}_{\text{direction}}, \{x_{\text{shadow}}, x_{\text{house}}\}\rangle$, $C_{\text{adjacent\_1}} = \langle \mu_{\text{adjacent}}, \{x_{\text{shadow}}, x_{\text{house}}\}\rangle$, $C_{\text{between}} = \langle \mu_{\text{between}}, \{x_{\text{pool}}, x_{\text{road}}, x_{\text{house}}\}\rangle$, $C_{\text{near}} = \langle \mu_{\text{near}}, \{x_{\text{house}}, x_{\text{pool}}\}\rangle$, $C_{\text{surrounds}} = \langle \mu_{\text{surrounds}}, \{x_{\text{garden}}, x_{\text{pool}}\}\rangle$, $C_{\text{adjacent\_2}} = \langle \mu_{\text{adjacent}}, \{x_{\text{house}}, x_{\text{garden}}\}\rangle$ and $C_{\text{aligned}} = \langle \mu_{\text{alig}}, \{x_{\text{house}}\}\rangle$. We do not consider the constraints representing the *near* relations between the elements of the aligned group, because they are already considered in the *alignment* relation.
- $\mathcal{D} = \{D_{\text{shadow}}, D_{\text{house}}, D_{\text{road}}, D_{\text{pool}}, D_{\text{garden}}, D_{\text{group\_houses}}\}$ where D_i represents the possible candidates for x_i. The regions in each D_i for $i \in \{\textit{shadow}, \textit{house}, \textit{road}, \textit{pool}, \textit{garden}\}$ are shown in Fig. 8.

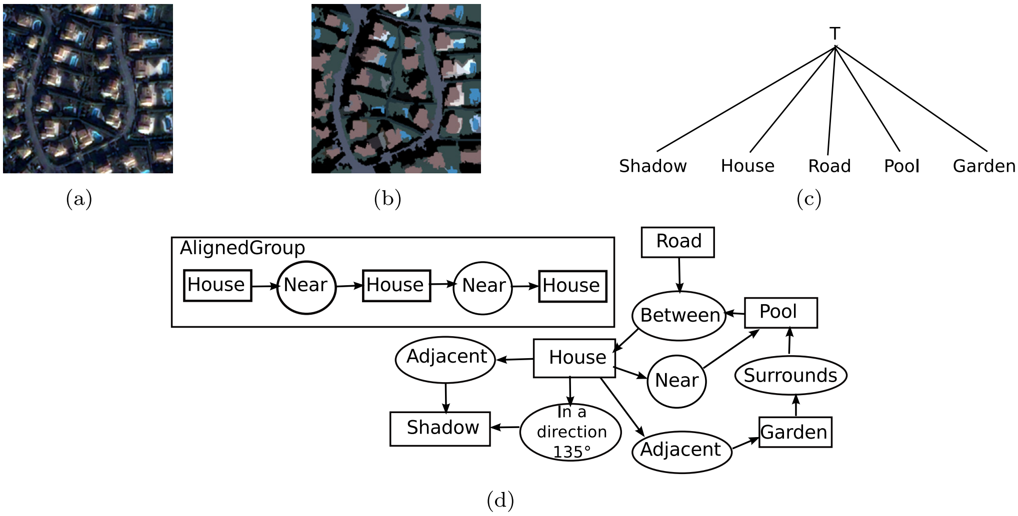
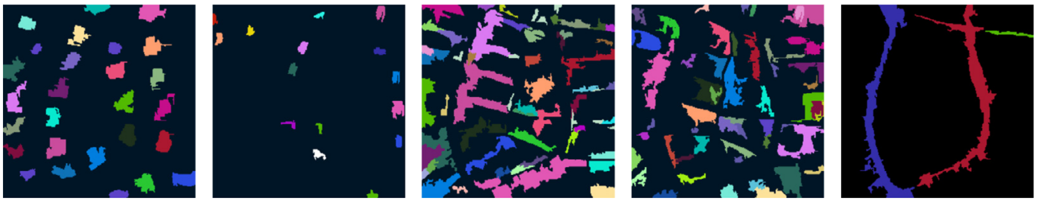


Fig. 7. Images and conceptual graph. (a) Example image. (b) Segmented and manually labeled image. The labels are: gardens in green, houses in orange, shadows in black, pools in blue, roads in gray. (c) Concept hierarchy. (d) Conceptual graph describing "the group of neighboring houses forming an aligned group which has a pool located in the garden at the "back" of the house, and which has a shadow". (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) D_{houses} (b) D_{pool} (c) D_{shadow} (d) D_{garden} (e) D_{road}

Fig. 8. Regions in the domains of each variable of the CSP problem representing the instantiations of the conceptual graph in Fig. 7(d) in the image shown in Fig. 7(b).

Table 1
Evolution of the fuzzy sets μ_{houses} when making D_{houses} arc-consistent at different iterations. Gray levels represent the values of μ_{houses} (white = 1, black = 0).

Iteration 0	Iteration 1	Iteration 4	Iteration 5	Iteration 15	Iteration 16	Iteration 38

Tables 1, 2, 3, 4 and 5 show the evolution of μ_{x_i} for $i \in \{house, shadow, pool, road, garden\}$, respectively. Table 6 shows the domain of $x_{\text{group\_houses}}$, we only show the iterations for $D_{\text{group\_houses}}$ when it has changed. The membership degree of a region to μ_{x_i} is represented by its gray value. The images representing the groups belonging to $D_{\text{group\_houses}}$ only show the elements of $D_{\text{group\_houses}}$, without the membership degrees. For illustrative purposes we evaluate the alignment constraint at the first iteration in order to show the evolution of $D_{\text{group\_houses}}$.

At the first iteration we check C_{aligned} and obtain 17 groups. At iteration 4 the constraint C_{between} is propagated and the domain of x_{houses} is modified. As D_{houses} has been modified the alignment condition is propagated again, and several groups are eliminated. At iteration 15 the constraint $C_{\text{adjacent\_2}}$ is propagated, and again the domain of x_{houses} is modified. Hence, at iteration 16 the C_{aligned} is propagated, resulting in the elimination of more groups and a modification in the domain of x_{houses}. At this iteration we observe the impact of the dual view of the C_{aligned}

Table 2
Evolution of the fuzzy sets μ_{shadow} when making D_{shadow} arc-consistent at different iterations. Gray levels represent the values of μ_{shadow}.

Iteration 0	Iteration 1	Iteration 4	Iteration 5	Iteration 15	Iteration 16	Iteration 38

Table 3
Evolution of the fuzzy sets μ_{pool} when making D_{pool} arc-consistent at different iterations. Gray levels represent the values of μ_{pool}.

Iteration 0	Iteration 1	Iteration 4	Iteration 5	Iteration 15	Iteration 16	Iteration 38

Table 4
Evolution of the fuzzy sets μ_{road} when making D_{road} arc-consistent at different iterations. Gray levels represent the values of μ_{road}.

Iteration 0	Iteration 1	Iteration 4	Iteration 5	Iteration 15	Iteration 16	Iteration 38

Table 5
Evolution of the fuzzy sets μ_{garden} when making each D_{garden} arc-consistent at different iterations. Gray levels represent the values of μ_{garden}.

Iteration 0	Iteration 1	Iteration 4	Iteration 5	Iteration 15	Iteration 16	Iteration 38

constraint, since it modifies both D_{houses} when it is considered as a relation and $D_{\text{group\_houses}}$ when it is considered as a variable. The final results are shown at iteration 38, where two groups are obtained: one group is contained in the other group, thus it is equivalent as having one only group. This example illustrates how the proposed algorithm is able to reduce the domains of the variables to obtain the regions in the image which satisfy the conceptual graph.

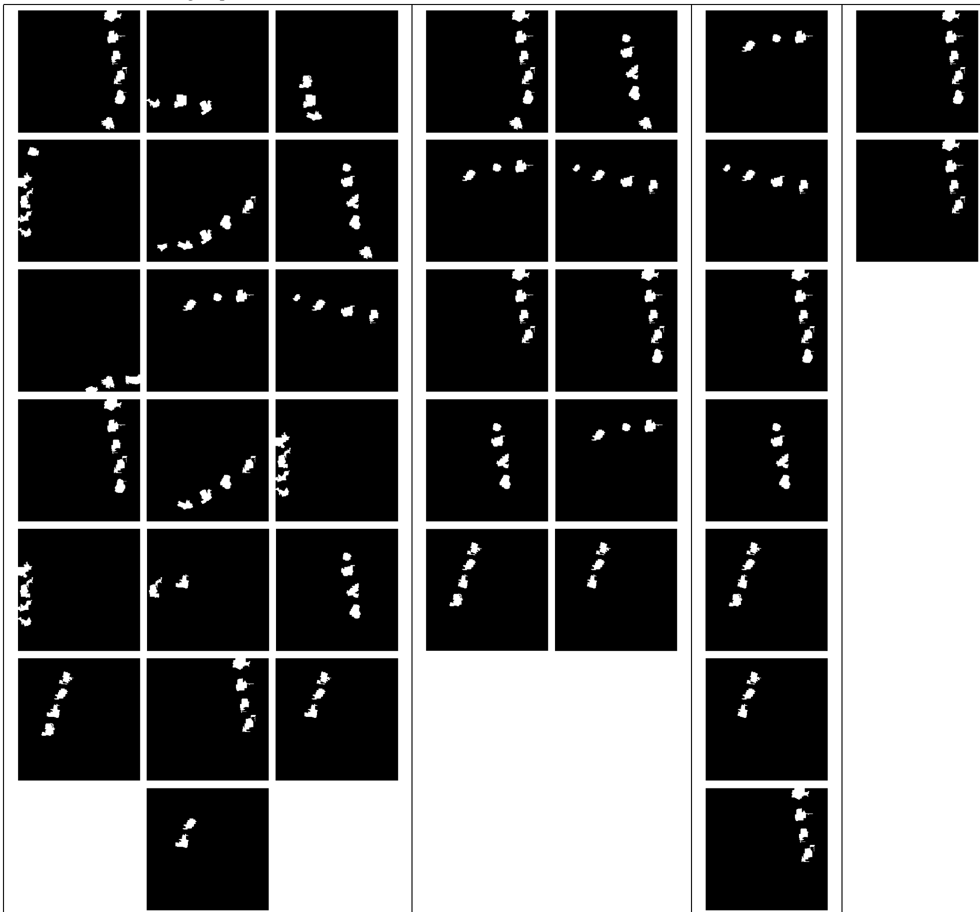
6. Interpreting an unlabeled image

When dealing with an unlabeled image, the difficulty lies in adequately creating the domain of regions which can represent each concept node of the model conceptual graph. As mentioned in Section 2.3, we tackle this difficulty by performing a multi-scale segmentation, which allows us to extract objects of different sizes. Moreover, the multi-scale segmentation provides an explicit hierarchical organization of the regions which can be useful for spatial reasoning. In our experiments, we used a hierarchical Mean Shift algorithm [14,36].

We assume that the regions obtained from the segmentation are object candidates. By considering this hypothesis we do not have to worry about segmentation problems such as a region containing two objects or an object split into several regions.

The membership functions of each domain of the FCSP are constructed over the set of regions obtained from the segmentation. To estimate the initial membership function we use two types of information:

Table 6
Objects belonging to D_{group} at different iterations in the process of solving the FCSP.



Iteration 1	Iteration 5	Iteration 16	Iteration 38

- the approximate size of objects represented by the concept nodes of the graph,
- the knowledge about the extraction of certain types of concepts of the vocabulary over which the conceptual graph is built.

For the first type of information, we assume that it is possible to know the typical sizes of the objects that we are searching, as in [8]. This information is given as linguistic terms {very small, small, medium, large, very large} which are modeled using trapezoidal membership functions over $\mathbb{R}$. The parameters defining these functions can be learned according to the scene. The second type of information depends on the concepts. In remote sensing, we can exploit the fact that we know how to extract certain classes of concepts, for instance:

- water (using Normalized Difference Water Index NDWI [30]),
- vegetation (using Normalized Difference Vegetation Index NDVI [22]),
- shadow (using a hysteresis threshold over the intensity image), etc.

Let $(T_C, \leq_C)$ be the concept hierarchy of the vocabulary over which the conceptual graph is built. Let H_C be a set representing the classes of concepts that we know how to extract. We propose to add the concepts of H_C to T_C, and also add a concept *"Other"* to represent all the concept classes which are not "a_kind_of" one of the concepts in H_C. For example Fig. 9(a) illustrates an initial concept hierarchy and Fig. 9(b) the augmented hierarchy.

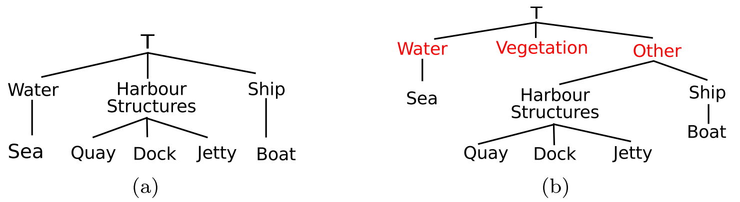


Fig. 9. (a) Initial T_C. (b) Augmented T_C using the concepts $H_C = \{water, vegetation\}$.

We construct the membership functions of the domains over the regions provided by the segmentation using the inclusion relation between the categories of T_C. First we extract in the image the regions corresponding to the classes of concepts in H_C. Then we compute the "*Other*" class as the complement of the disjunction of the known classes. Suppose that we can define every $c \in H_C \cup \{\text{"}Other\text{"}\}$ as a fuzzy set μ_c over the image space. Then for every concept node in the conceptual graph, represented as a variable x_i, we construct its membership function over the regions of the image as:

$$\mu_{x_i}(v) = \left[\bigwedge_{\{c \in H_C \mid \text{type}(x_i) \leq_C c\}} F(\mu_c, v) \right] \wedge \mu_{\text{size\_i}}(v) \tag{10}$$

where $\mu_{\text{size\_i}}$ represents the membership function corresponding to the size of the objects represented by x_i, and F is a comparison measure which evaluates how well v matches with μ_c. For instance, F can be a mean measure. The first term in Eq. (10) is the conjunction of all the membership degrees of the classes of H_C for which the type of the concept x_i is a sub-category. Adding the *"Other"* class allows us to have an initial membership degree for each class which already excludes the other classes. Another alternative would be to give a membership degree equal to 1 to all the regions for which we do not have any information.

Once the initial membership functions for the variables that represent the concept nodes of the model are estimated, we can apply Algorithm 3 to find the arc-consistent domains. The proposed method is illustrated in Fig. 1.

6.1. Finding a solution

Applying the arc-consistency algorithm reduces the domain of search of the homomorphism (expressed as a FCSP). The instantiations of the model, or the solutions of the FCSP, are obtained by searching in the reduced domains using a branch and bound algorithm as in [17,31]. The regions of the image are then labeled according to the instantiations of the model. When labeling the regions, it is possible to encounter a conflict among two instantiations. A conflict between two instantiations arises when there is a region in an image that is labeled differently according to each instantiation. In case of a conflict we propose to eliminate the instantiation which is less important, according to some order.

There are several strategies to order the solutions obtained from a FCSP [18]. We propose to use an egalitarist approach [3], where the order is given by the least satisfied relation of the FCSP. An example of this type of approach is the maximin strategy: let V_1 and V_2 be two tuplets containing an instantiation of the model, we say that $V_1 \leq V_2$ if and only if $\text{cons}(V_1) \leq_{\text{max-min}} \text{cons}(V_2)$. The egalitarist approach, in particular the maximin, preserves a basic property of FCSP, where a solution violating a constraint is not considered as feasible. However, one drawback of the maximin approach is that it is not capable of discriminating among solutions having different satisfaction degrees for each constraint except for the minimum degree [18,19]. To refine the ordering obtained by the maximin strategy, the authors in [3] propose to use a maxisum approach to discriminate between the solutions that cannot be discriminated using the maximin approach, and in [18,20,32,39] a lexicographical ordering is used to improve the discrimination of the maximin. Here we also use the leximin ordering as in [18,20,32,39] to define an order between solutions. Given two vectors $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$, assume that $A^* = (a_1^*, \ldots, a_n^*)$ and $B^* = (b_1^*, \ldots, b_n^*)$ contain the elements of A and B in ascending order, respectively. We say that A is less than B according to the leximin order if and only if there exists $k \leq n$ such that $a_k^* \leq b_k^*$ and $a_j^* = b_j^*$ for all $j < k$. The leximin order is applied using the original membership degrees defined over the regions before the arc-consistency algorithm was applied.

In our experiments we did not have a great number of conflicting instantiations. However if the number of conflicting instantiations is very high then the ordering of the solutions according to the leximin can lead to an exponential

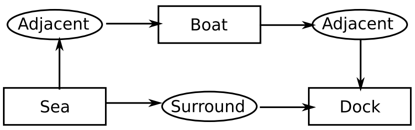
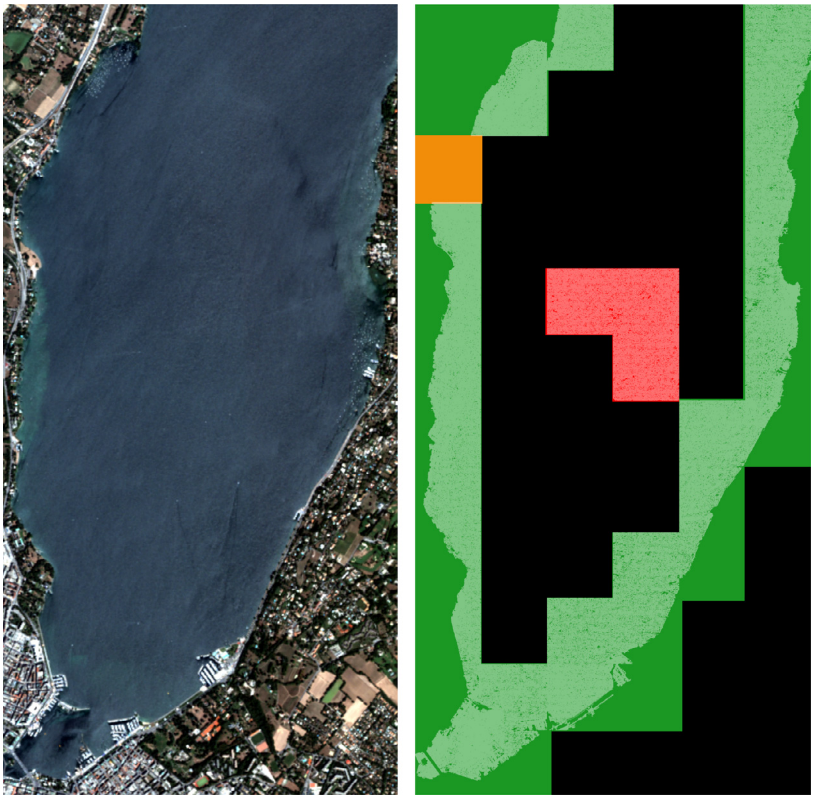


Fig. 10. Conceptual graph used for the interpretation of Fig. 11(a). This graph was built over the vocabulary from Fig. 11(b).



(a) (b)

Fig. 11. (a) Quickbird image of a lake. (b) Image containing the objects detected as instantiations of the model. The green regions represent the true positive tiles where the model was correctly identified, at the interior of each tile the detected objects are represented in lighter green. The orange region represents a false negative tile where the algorithm did not detect the model. The red regions represent the false positive tiles where the algorithm incorrectly detected the model. And the black regions represent the true negative tiles where the model was correctly not detected. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

number of optimal solutions. To alleviate this problem it is possible to enforce arc-consistency based on the leximin criterion as proposed in [12]. Using this type of arc-consistency would enforce the leximin ordering while reducing the domains. Moreover it is possible to improve the performance of the arc-consistency algorithm using substitution techniques as in [28] or [11] where a method for arc-consistency based in the leximin criterion is proposed.

6.2. Results

We illustrate the proposed method in two situations: searching for the harbors in an image and interpretating an image containing an airport, in two Quickbird images with a resolution of 0.7 m.

Finding harbors We applied the method summarized in Fig. 1 to the image displayed in Fig. 11(a) to extract the harbors in the image. The conceptual graph that we used to represent the structure of a harbor is given in Fig. 10. In this example, we do not use alignments because of the low quality of the segmentation of boats.

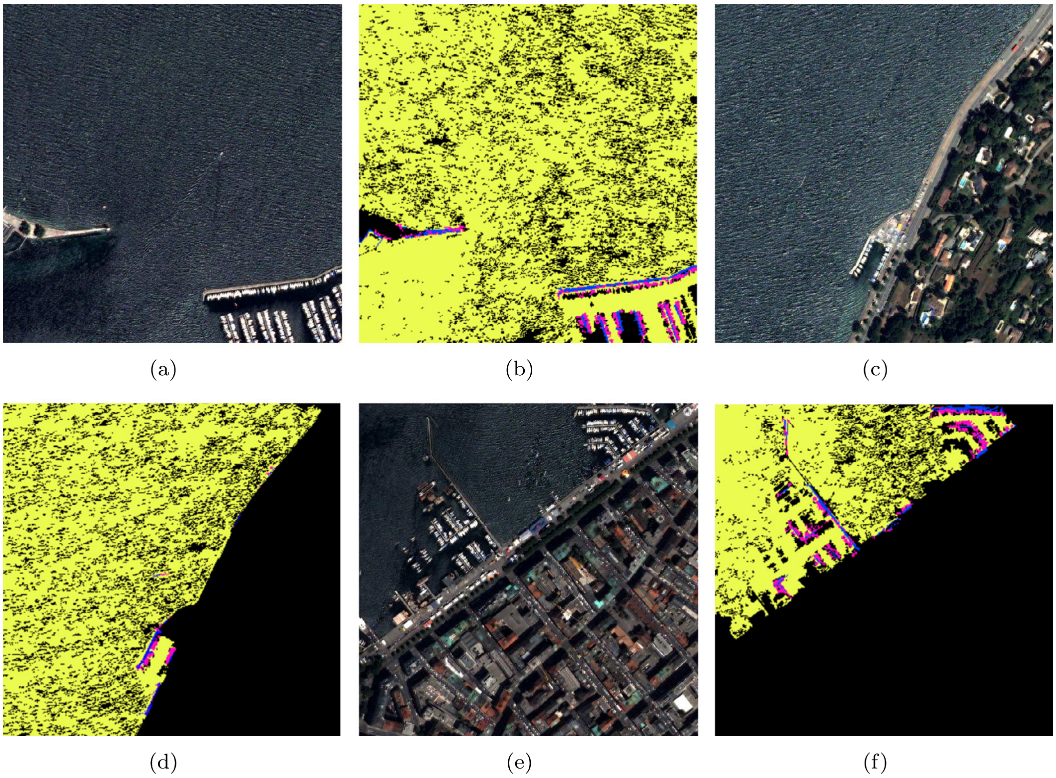


Fig. 12. Examples of instantiations of the model of Fig. 10 in the image tiles of Fig. 11(a). Figures (a), (c) and (e) correspond to the original tile, and figures (b), (d) and (f) to the model instantiations, respectively. In yellow the sea, in pink the boats and in blue the docks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The image in Fig. 11(a) has 5901 × 11 801 pixels. We divided the image into 72 tiles of approximately 985 × 985 pixels, and applied the method in each tile. To extract the initial candidate regions for each domain, the set H_C containing the known classes was {*water*, *vegetation*}, and in addition we used the fact that docks have a linear structure.[3] Fig. 9(b) illustrates the concept hierarchy.

From the 72 tiles, we correctly detected the presence and no-presence of a harbor in 68 tiles. In Fig. 11(b) we show the true positive, true negative, false positive and false negative tiles. We can see that although we used a very simple graph it was possible to extract the zones of the image which correspond to the harbors. In Fig. 12 we show some examples of the instantiations of the model. We can see that in (a) to (f) the harbors are correctly detected. Even if the harbor of (c) is very small, the method finds satisfactory instantiations in (d). Most of the correct instantiations have a higher consistency value (Eq. (3)) than the false detections. This example illustrates how the method reduces considerably the regions of the image where we can search for a harbor.

Interpretation of an airport image The second example addresses the problem of interpretating the airport image of Fig. 13(a). For this example we used the model in Fig. 14(a) with the concept hierarchy in Fig. 14(b).

The results of the multiscale segmentation are shown in Figs. 13(b), (c) and (d). The instantiations of the model are shown in Fig. 13(e). We can see that although the segmentation produces more than 1000 regions, the instantiations coincide with the airport. There is only one building which was not detected, because it was split into two regions in the segmentation and one of the regions satisfies the condition of being adjacent to its shadow, but they do not satisfy the condition of being adjacent to the concrete surface.

[3] To evaluate if a region is linear we computed the ratio of its largest principal moment by the smallest principal moment, and we considered that an object was linear if this ratio was equal or greater than 4. This value was set experimentally.

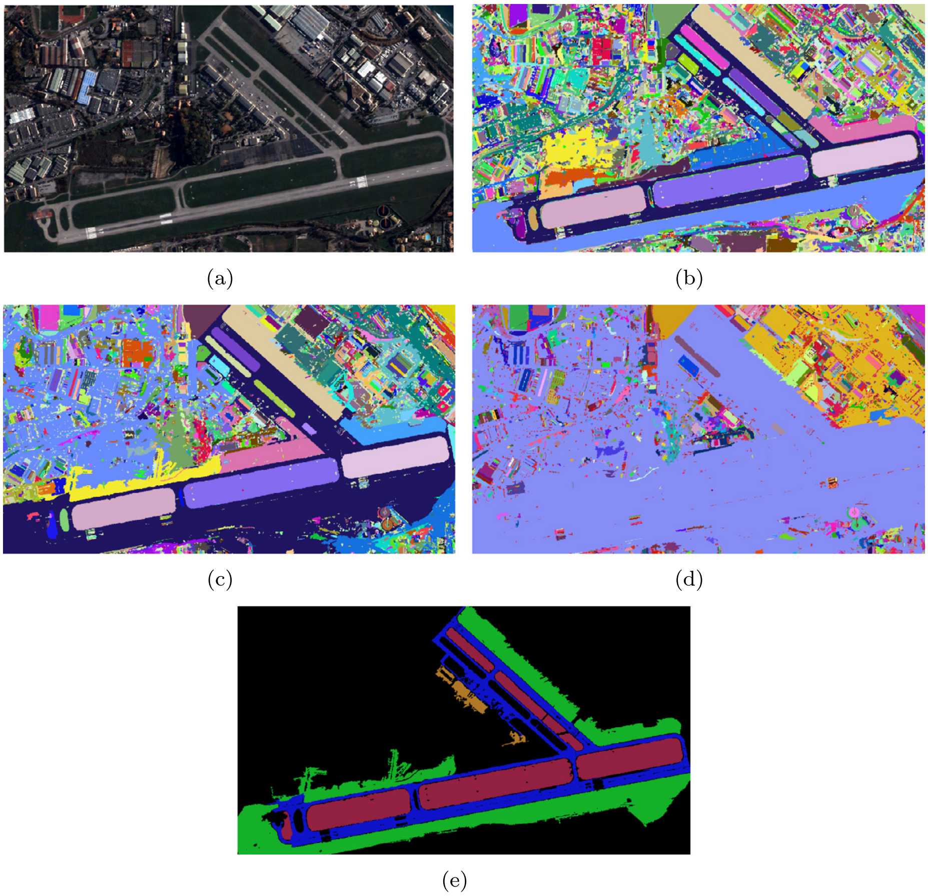
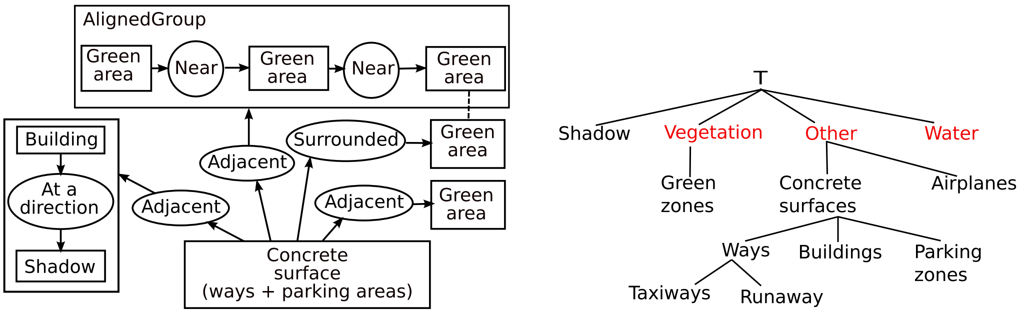


Fig. 13. (a) Airport image. (b), (c) and (d) Results of the multiscale segmentation at 3 different scales. (e) Instantiations of the conceptual graph of Fig. 14(a) in the image of figure (a). In red the aligned green zones, in green the green zone which does not belong to the aligned group, in blue the concrete area, and in yellow the building and its shadow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Conceptual graph (b) Concept hierarchy

Fig. 14. Knowledge used for the interpretation of Fig. 13(a).

Discussion Through these two examples we have shown the interest of the proposed approach for obtaining the instantiations of complex objects in an image. Although the segmentations in both examples were not perfect, the methodology was able to detect the instantiations of the models in the images. Moreover, we were able to extract the instantiations of the complex objects without needing to specify a method for extracting each object of the scene.

The use of homomorphisms between the model conceptual graph and the image allows us to retrieve several instantiations of the objects composing complex objects. For instance we were able to retrieve the two groups of aligned green zones in the airport, or different boats which are adjacent to a dock in the harbors. Moreover, using the conceptual graphs for the representation of the desired structure allows us to represent groups having a relation with other objects and members of the group having relations with objects outside the group. The richness of this representation leads to the description of a complex structure like an airport in a simple way.

In these two examples we used very simple concept hierarchies and only few classes H_C to help us to identify the initial candidates for the instantiations of the concept nodes in the image. However, if H_C has more elements it is possible to have a smaller set for the initial candidates and the number of false detections can be reduced.

7. Conclusion

We have addressed the problem of incorporating complex spatial relations in a model which represents a scene that we want to find in an image. For this, we first adapted a representation scheme to introduce this type of information in a model, then we addressed the problem of identifying the model in the image, by simultaneously considering some of the possible information imperfections which are present in this type of problem.

The mapping problem was represented as a graph homomorphism which allows us to have several instantiations of a model in an image. This flexibility is adequate for satellite images since most of the time the number of instantiations of a model in an image is unknown. This is an important and new characteristic of the proposed approach.

The problem of obtaining the graph homomorphism in an image was formulated as a CSP, as in [7]. However, due to the imprecision of the spatial relations, it was necessary to move to a more flexible formalism such as FCSP. We adapted the algorithm to deal with groups of objects which can be aligned or not. This extension is a new contribution.

Finally, we proposed a methodology to find the instantiations of a conceptual graph in an unlabeled image. Our method was successfully applied to unlabeled images obtaining good results, even if there were segmentation errors. The results demonstrate the interest of using the spatial relations for the interpretation of images.

This work could also be used for different applications, such as in the medical domain, extending the work in [34] (using fuzzy relations but with crisp CSP) for instance. Different relations would be relevant but in general the proposed methodology can be applied. The extension of FCSP to groups of objects can be useful to represent pathologies that can occur at different locations (e.g. multiple sclerosis, groups of microcalcifications…).

References

[1] J. Atif, C. Hudelot, G. Fouquier, I. Bloch, E.D. Angelini, From generic knowledge to specific reasoning for medical image interpretation using graph based representations, in: International Joint Conference on Artificial Intelligence IJCAI'07, 2007, pp. 224–229.
[2] U. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, M. Heynen, Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information, ISPRS J. Photogramm. Remote Sens. 58 (3–4) (2004) 239–258.
[3] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, H. Fargier, Semiring-based CSPs and valued CSPs: frameworks, properties, and comparison, Constraints 4 (3) (1999) 199–240.
[4] I. Bloch, Fuzzy spatial relationships for image processing and interpretation: a review, Image Vis. Comput. 23 (2005) 89–110.
[5] I. Bloch, O. Colliot, R. Cesar, On the ternary spatial relation between, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 36 (2) (2006) 312–327.
[6] I. Bloch, T. Géraud, H. Maître, Representation and fusion of heterogeneous fuzzy information in the 3D space for model-based structural recognition—application to 3D brain imaging, Artif. Intell. 148 (1–2) (2003) 141–175.
[7] M. Chein, M.-L. Mugnier, Graph-Based Knowledge Representation and Reasoning—Computational Foundations Conceptual Graphs, Advanced Information and Knowledge Processing, Springer, 2008.
[8] M. Ciucu, P. Héas, M. Datcu, J.C. Tilton, Scale space exploration for mining image information content, in: Mining Multimedia and Complex Data, KDD Workshop MDM/KDD 2002, PAKDD Workshop KDMCD 2002, Revised Papers, Springer, 2002, pp. 118–133.
[9] O. Colliot, O. Camara, I. Bloch, Integration of fuzzy spatial relations in deformable models—application to brain MRI segmentation, Pattern Recognit. 39 (8) (2006) 1401–1414.
[10] M. Cooper, T. Schiex, Arc consistency for soft constraints, Artif. Intell. 154 (1) (2004) 199–227.
[11] M.C. Cooper, Reduction operations in fuzzy or valued constraint satisfaction, Fuzzy Sets Syst. 134 (3) (2003) 311–342.
[12] M.C. Cooper, S. de Givry, M. Sánchez, T. Schiex, M. Zytnicki, T. Werner, Soft arc consistency revisited, Artif. Intell. 174 (7) (2010) 449–478.
[13] D. Crevier, R. Lepage, Knowledge-based image understanding systems: a survey, Comput. Vis. Image Underst. 67 (2) (1997) 161–185.
[14] D. DeMenthon, R. Megret, Spatio-temporal segmentation of video by hierarchical mean shift analysis, in: Statistical Methods in Video Processing Workshop, 2002.
[15] A. Deruyver, Y. Hodé, Constraint satisfaction problem with bilevel constraint: application to interpretation of over-segmented images, Artif. Intell. 93 (1–2) (1997) 321–335.

[16] A. Deruyver, Y. Hodé, L. Brun, Image interpretation with a conceptual graph: labeling over-segmented images and detection of unexpected objects, Artif. Intell. 173 (14) (2009) 1245–1265.
[17] D. Dubois, H. Fargier, H. Prade, Possibility theory in constraint satisfaction problems: handling priority, preference and uncertainty, Appl. Intell. 6 (4) (1996) 287–309.
[18] D. Dubois, H. Fargier, H. Prade, Refinements of the maximin approach to decision-making in a fuzzy environment, Fuzzy Sets Syst. 81 (1) (1996) 103–122.
[19] D. Dubois, P. Fortemps, Computing improved optimal solutions to max–min flexible constraint satisfaction problems, Eur. J. Oper. Res. 118 (1) (1999) 95–126.
[20] H. Fargier, Modèles et algorithmes pour l'aide à la décision, Université Paul Sabatier, Toulouse, France, 2006, Habilitation à diriger des recherches.
[21] G. Fouquier, J. Atif, I. Bloch, Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations, Comput. Vis. Image Underst. 116 (2012) 146–165.
[22] S. Goward, B. Markham, D. Dye, W. Dulaney, J. Yang, Normalized difference vegetation index measurements from the Advanced Very High Resolution Radiometer, Remote Sens. Environ. 35 (2–3) (1991) 257–277.
[23] D. Guo, H. Xiong, V. Atluri, N. Adam, Object discovery in high-resolution remote sensing images: a semantic perspective, Knowl. Inf. Syst. 19 (2) (2009) 211–233.
[24] J. Guo, H. Zhou, C. Zhu, Cascaded classification of high resolution remote sensing images using multiple contexts, Inf. Sci. 221 (2013) 84–97.
[25] C. Hudelot, J. Atif, I. Bloch, Fuzzy spatial relation ontology for image interpretation, Fuzzy Sets Syst. 159 (15) (2008) 1929–1951.
[26] C. Hudelot, N. Maillot, M. Thonnat, Symbol grounding for semantic image interpretation: from image data to semantics, in: Tenth IEEE International Conference on Computer Vision Workshops, 2005, ICCVW'05, China, Beijing, 2005.
[27] F. Le Ber, J. Lieber, A. Napoli, Les systèmes à base de connaissances, in: J. Akoka, I. Comyn Wattiau (Eds.), Encyclopédie de l'informatique et des systèmes d'information, Vuibert, 2006, pp. 1197–1208.
[28] C. Lecoutre, O. Roussel, D. Dehani, WCSP Integration of soft neighborhood substitutability, in: M. Milano (Ed.), Principles and Practice of Constraint Programming, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2012, pp. 406–421.
[29] Y. Liu, Y. Zhanga, Y. Gaoa Gnet, A generalized network model and its applications in qualitative spatial reasoning, Inf. Sci. 178 (2008) 2163–2175.
[30] S. McFeeters, The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features, Int. J. Remote Sens. 17 (7) (1996) 1425–1432.
[31] J. Metzger, F. Le Ber, A. Napoli, Modeling and representing structures for analyzing spatial organization in agronomy, in: Conceptual Structures for Knowledge Creation and Communication, in: A. de Moor, W. Lex, B. Ganter (Eds.), 11th International Conference on Conceptual Structures, ICCS 2003, vol. 2746, Springer, Dresden, Germany, 2003, pp. 215–228.
[32] R. Möller, T.H. Näth, Implementing probabilistic description logics: an application to image interpretation, in: A.G. Cohn, D.C. Hogg, R. Möller, B. Neumann (Eds.), Logic and Probability for Scene Interpretation, Dagstuhl Seminar Proceedings, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany, 2008, pp. 1862–4405.
[33] S.W. Myint, P. Gober, A. Brazel, S. Grossman-Clarke, Q. Weng, Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery, Remote Sens. Environ. 115 (5) (2011) 1145–1161.
[34] O. Nempont, J. Atif, I. Bloch, A constraint propagation approach to structural model based image segmentation and recognition, Inf. Sci. 246 (2013) 1–27.
[35] G.T. Papadopoulos, C. Saathoff, H. Escalante, V. Mezaris, I. Kompatsiaris, M. Strintzis, A comparative study of object-level spatial context techniques for semantic image analysis, Comput. Vis. Image Underst. 115 (9) (2011) 1288–1307.
[36] S. Paris, F. Durand, A topological approach to hierarchical segmentation using mean shift, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR07, 2007, pp. 1–8.
[37] A. Perchant, I. Bloch, Fuzzy morphisms between graphs, Fuzzy Sets Syst. 128 (2) (2002) 149–168.
[38] A. Rosenfeld, R. Hummel, S. Zucker, Scene labeling by relaxation operations, IEEE Trans. Syst. Man Cybern. 6 (6) (1976) 420–433.
[39] C. Saathoff, S. Staab, Exploiting spatial context in image region labelling using fuzzy constraint reasoning, in: WIAMIS '08: Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, Washington, DC, USA, 2008, pp. 16–19.
[40] I. Sebari, D.-C. He, Automatic fuzzy object-based analysis of VHSR images for urban objects extraction, ISPRS J. Photogramm. Remote Sens. 79 (2013) 171–184.
[41] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.
[42] J.F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, 1984.
[43] A. Sowmya, J. Trinder, Modelling and representation issues in automated feature extraction from aerial and satellite images, ISPRS J. Photogramm. Remote Sens. 55 (1) (2000) 34–47.
[44] M.-C. Vanegas, Spatial relations and spatial reasoning for the interpretation of earth observation images using a structural model. Ph.D. thesis, Télécom ParisTech, 2011.
[45] M.-C. Vanegas, I. Bloch, J. Inglada, A fuzzy definition of the spatial relation "surround"—application to complex shapes, in: EUSFLAT, 2011, 2011, pp. 844–851.
[46] M.-C. Vanegas, I. Bloch, J. Inglada, Alignment and parallelism for the description of high resolution remote sensing images, IEEE Trans. Geosci. Remote Sens. 51 (6) (2013) 3542–3557.
[47] J. Yuan, J. Li, B. Zhang, Exploiting spatial context constraints for automatic image region annotation, in: Proceedings of the 15th International Conference on Multimedia, ACM, 2007, pp. 595–604.