An Ontology-Based Model for Representing Image Processing Objectives

Régis Clouard, Arnaud Renouf, Marinette Revenu

Electronic version of an article published as International Journal of Pattern Recognition and Artificial Intelligence, Vol. 24, No. 8 (2010), 1181-1208.
DOI No: 10.1142/S0218001410008354

An Ontology-Based Model for Representing Image Processing Application Objectives

Régis CLOUARD, Arnaud RENOUF, Marinette REVENU

Abstract

This paper investigates what kinds of information are necessary and sufficient to design and evaluate image processing software programs and proposes a representation of these information elements using a computational language performable by vision systems and understandable by experts. The language is built upon a formulation model which distinguishes the specification of a goal and the definition of an input image class. Goals are stated in terms of tasks together with result samples. Image classes are defined by both linguistic and iconic descriptions. The model is implemented as an OWL domain ontology which provides the primitives for the formulation language.

***Index Terms*–Ontology Design, Knowledge Acquisition, Image Processing and Computer Vision.**

1 Introduction

Explicit representation of objectives is of paramount importance when developing image processing applications. An image processing application is a software program tailored to a given image transformation goal and a given input image class. Under a restricted definition of image processing, goals refer only to image-to-image transformations without interpretation of the

image content. Six categories of goals are thus considered: image restoration, image enhancement, image compression, image reconstruction, image segmentation, and object detection. An image class is a set of images that share many features, on condition that these features are meaningful for the application.

The work described hereafter does not address the design of image processing algorithm, but the use of existing algorithms for developing image processing applications. The representation of image processing objectives contains essential pieces of information that are required at each level of the application development process. Since each image processing algorithm has its own domain of applicability, the solution design should find in the representation of the objective rationales to choose relevant algorithms and tune their parameters. Furthermore, since evaluation has meaning only in reference to a goal [49], the objective is necessary to define appropriate evaluation metrics for each application.

An image processing application is one component of the low level part of a more global image analysis and computer vision system. Its role is completely determined by the high-level part which uses its results to perform interpretation, visualization, storage, or transmission of the image data. But, even if an image processing application is only one element of an imagery system, it is often one of its bottlenecks. Consider an image analysis application which addresses the automation of a diachronic analysis of long-term agricultural landscape changes. A same landscape sector is compared year by year, always at the same season, in order to quantify evolution of cultivated surface areas. The image class is composed of color satellite images of landscape with the same resolution (see example Fig. 1). The objective of the image processing part is to segment the input images in order to isolate each potential vegetation area into a unique region. The resulting regions will then be fed into a classifier which is trained to identify various categories of geographic objects: field, forest, hedge, city, etc. The performance of the classifier clearly depends on the accuracy of the image segmentation process.

The development of such an image processing application cannot be done without an explicit representation of the objective because the problem data are not entirely included in the input image data. Two reasons can account for that. First, image data are intrinsically incomplete, degraded and corrupted. The image acquisition process is responsible for creating images that are underconstrained representations of the scene[1] because it causes loss of information (*e.g.*, third dimension, motion), mixture of several factors in the

[1]The term "scene" refers to the observed or calculated phenomenon, whether it is natural or artificial.

value of a pixel (*e.g.*, texture, lighting, geometry), introduction of false values (*e.g.*, noise, chromatic aberration), and alteration of the original information (*e.g.*, geometric distortion, blurring) [14]. Second, image content does not make sense by itself. An image is inherently ambiguous and does not provide information about its content. Without a subject matter, an image does not allow to make a distinction between relevant and irrelevant information. What is relevant for one application may be irrelevant for another. Furthermore, there exists no intrinsic relevant information. For example, apparently simple information such as object edges is difficult to accurately extract without knowledge about the scene. Edges are usually modeled as sharp changes in image intensity, but this is also the case for noise, shadows, and texture elements. Accordingly, the problem to be solved is held by sources external to the input images and must be formulated.

Figure 1: The image processing objective of the aerial imagery application is to segment each vegetation area into a region.

In this paper, we propose a computational language for explicitly representing image processing objectives. We argue that it is possible to define such a language that is independent from the application domains, but that allows the representation of knowledge about the application domains. The study is restricted to the case of still image processing; however, the model is readily extensible to support image sequence processing. The paper is organized as follows: after a review of various approaches to image processing objective formulation in Section 2, a model that is the support of our formulation language is proposed in Section 3. An implementation of this model

in the form of a domain ontology is detailed in Section 4, and two examples of the use of the language are discussed in Section 5. Finally, findings and future prospects of this work are presented in Section 6.

2 Objective Formulation in Image Processing

Three categories of a priori information need to be considered for an explicit formulation of an image processing objective:

- The expression of the aim of the application in order to give a subject matter;
- The description of the image acquisition process in order to recover the lost, altered, and hidden information about the scene;
- The definition of the application domain concepts in order to assign a semantics to the scene by identifying information to be considered relevant.

The first category of information corresponds to the definition of the *goal*, and the last two categories compose the definition of the *image class*. The specification of these pieces of information is regarded as a particularly complex activity since it is of *qualitative nature*. This implies that no exhaustive or exact specification of the application objective can be reached and one has to resort to only an approximate description [40]. This is the consequence of the so-called sensory and the semantic gaps [45] (see Fig. 2). The sensory gap is the difference between real objects and their representation in images, whereas the semantic gap is the difference between the interpretation of the object that one can make from an image content and from a description of the object in terms of low-level image features. Hence, formulating means bridging the sensory and the semantic gaps in order to represent a real-world objective with measurable symbolic features used to process images.

2.1 Related Work

In this section, the analysis of various image processing systems that address the formulation of user objectives is done along the two axes: the image class representation and the goal representation. It shows how the different types of information representation affect the expressive power of the formulation.

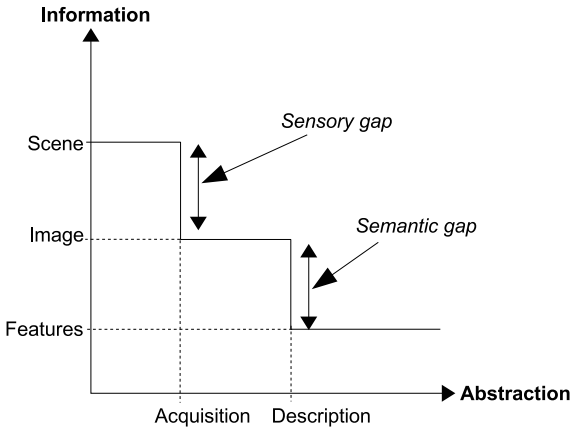


Figure 2: The sensory gap is the gap between the scene in the world and its representation in an image and the semantic gap is the gap between an image representation and its description with low level features.

2.1.1 Image Class Definition

The image class definition can either be done by *extension* by means of sample images or by *intension* through a linguistic description.

Definition by extension. The a priori information is represented by sample image parts. Two types of sample image parts have been considered: blobs and patches.

- A blob represents a region in the image which delineates one object of interest (*e.g.*, Fig. 3.a) or an image area. Blobs can be drawn manually or obtained from an interactive segmentation [24, 3]. Information about the image class is then automatically extracted from these blobs. For example, the description of the objects of interest can be made thanks to color, size, and shape features extracted from the blobs. In the same way the noise features can be computed from an image area that is supposed to be uniform in the scene.

- A patch is a thumbnail that isolates one salient part of one object of interest. Thus, an object of interest is described by a series of patches with their spatial relations (*e.g.*, Fig. 3.b). Generally, patches are automatically extracted from sample images around points of interest [25, 1], considered as the most information-dense areas.

Definition by intension. The a priori information is represented by a linguistic description. The description language is generally built over a *domain ontology* which provides the representation primitives (*e.g.*, [23, 3, 29,

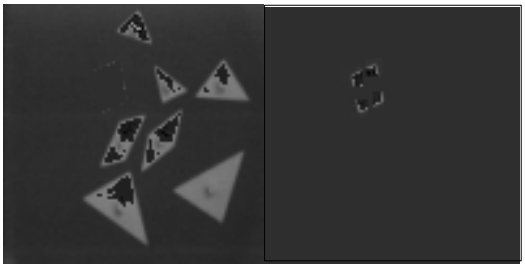


Figure 3: Two different ways for defining a tangram piece by extension: (a) with a blob (b) with patches.

19]. Thus, the description of an image class is an *application ontology*, which is made by choosing and reifying primitives of the domain ontology [5]. For example, N. Maillot et al. [29] propose the "Ontology of Visual Concepts" that defines the concepts of texture, color, geometry, and topology relationship. Fig. 4 gives a textual representation of an example of the description of a pollen grain made with concepts of this ontology. To better account for the variability of the visual appearance of objects, linguistic variables or fuzzy values can be used [33, 21] (*e.g.*, "pink", "circular", "slightly oblong", "is front of", "close to"). But quantitative values are required to build concrete applications. Therefore, intensional definition should also cope with the *symbol anchoring problem* [11] to connect the linguistic symbol to numerical image data. Symbol grounding can be carried out using dictionaries such as the Color Naming System [2] where each terms have been assigned to a predefined range of numerical values; this principle can also be extended to texture concepts with the Texture Naming System dictionary [38]. But, more often, symbol grounding is addressed as a learning problem from a base of blobs drawn on sample images [29, 28, 21].

Figure 4: Example of the definition of a Poaceae (Gramineae) pollen grain made with concepts of the 'Ontology of Visual Concepts' proposed by Maillot et al. [29].

Both approaches have their own strengths and weaknesses. The advantage of the definition by extension is to minimize the amount of a priori

information that it is necessary to supply. It reduces the cognitive load of the users since the formulation does not require any representation language (even if it may be tedious when the number of reference images to provide is high). The drawback is that the actual definition of the image class is implicitly done by the system from the sample images, generally through the use of predefined feature extraction or patch selection algorithms. This means that a part of the definition is assigned by the system and cannot be adapted to suit each individual application. On the contrary, the advantage of the definition by intension is to better reflect the user expertise about the scene. It provides a language able to represent the semantics of the scene and, thereby, able to capture the variability of the input images. In addition, this approach is applicable even for image sequences [12]. However, the drawback is that the construction of a linguistic description is known to be arduous [45].

2.1.2 Goal Specification

A goal can be formulated either as 'what to do' by means of a *task statement,* or 'what to get' through *result examples.*

Specification by task. A task describes a concrete processing objective. Constraints may be associated with the task to fine-tune its scope. For example, a request for the MVP system is '*radiometric correction*' [8] and a request for the LLVE system is '*find a rectangle whose area size is between 100 and 200 pixels*' [32].

Specification by example. A goal is formulated through one or more reference images which contain the representation of the results to be obtained from sample images. Three different representations can be found in the literature:

- Reference images as *sketches* drawn on test images (*e.g.*, Fig. 5.a). They specify the expected results as samples of contours [20] or regions [13, 27] to be considered.

- Reference images as *ground truth* for representative images [31] (*e.g.*, Fig.5.b). They define the exact results to be obtained.

- Reference images as rough *scribbles* drawn on test images (*e.g.*, Fig. 5.c) [26, 37]. They indicate the regions of interest as markers inside the regions (and the complementary regions.)

The two approaches are complementary. The specification by task has the advantage to span all image processing objectives [9]. Besides, the tasks can

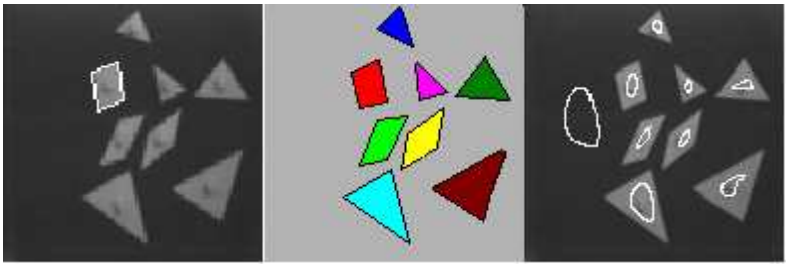


Figure 5: Three approaches for the specification by example of the tangram piece extraction objective: (a) sketch (b) manual segmentation (c) scribble.

take into account specific user requirements through associated constraints. However, the drawback is that the formulation is qualitative with no real link with the image data. This fact has two important consequences: first, the specification of a task is not explicit enough to predict the exact expected result, and second, there is only a finite number of possible goal representations with a predefined vocabulary. The advantage of the specification by example is that the formulation is quantitative and that it gets its values directly from the image data. Therefore, it allows for an infinite variety of goal representations. Moreover, it reduces the cognitive load of the users because a specialized formal vocabulary is not needed to state the problem. The drawback of this second approach is that a reference image is not sufficient to formulate all image processing objectives for at least three reasons. First, only object detection, image segmentation, and image enhancement goals can be addressed. Second, it does not cover all image classes. In particular, it gets tedious using it for 3D image or image sequence applications (though the sketch-based approach was extended to 3D images for a particular medical application [50]). Third, it does not provide a means to express some specific requirements. For instance, constraints such as "*prefer false detection to misdetection*" or "*prefer no result to bad result*" cannot be explicitly specified through reference images.

2.2 Conclusion

Having investigated the advantages and drawbacks of each these approaches, it clearly appears that a complete and generic formulation model should integrate the whole. The specification of a processing goal should be done by a task statement and grounded in image data by result examples. The definition of an image class should be done by intension in order to capture the semantics of the scene and by extension to get closer to the actual data and their internal consistency.

3 Objective Representation Model

In this section, the kinds of information we consider necessary and sufficient for designing and evaluating image processing applications are exhibited. A model is thus developed that provides a conceptualization of the objective formulation.

3.1 Image Class Definition

3.1.1 Phenomenological Hypothesis

Among the pieces of information that can be used to define an image class, we argue that only phenomenological information is necessary and sufficient [42]. Phenomenological information reflects a visual manifestation of the scene. Therefore, the purpose of formulation is not to describe the scene in its physical reality but only in the way it is perceived through images. This hypothesis has two strong consequences:

- The advantage is that the image class definition can be reduced to a denotation from visual cues. Thereby, it is not necessary to represent ontological knowledge about the application domain. For example, the real object "bus" in the aerial image Fig. 6.a can be reduced from the phenomenological point of view to a simple small white rectangle. This contributes to the existence of a core of information specific to image processing and independent from the application domain, upon which our model is built.

- The drawback is that it may be necessary to distinguish as many denotations of a real object as it presents distinct appearances in images. For example, consider the case of a bottle detection objective. Two distinct objects in the image class definition ought to be considered if there does not exist a common denotation between the two visual appearances of the bottle in Fig. 6.b and in Fig. 6.c.

3.1.2 A 3-Level Definition

For a complete definition of an image class, three different levels should to be considered: (1) the physical (2) the perceptive, (3) and the semantic levels [42] (see Fig. 7).

1. The *physical level* is the level of the measured signal that encodes the image data. The phenomenological hypothesis leads to describe this

Figure 6: Illustrations of the phenomenological hypothesis. (a) In this aerial image of a parking lot, the concept of bus can be reduced to a simple white homogeneous rectangular object of interest. The difference of appearance of a same bottle between image (b) and image (c) leads to consider them as two distinct objects of interest. *Images 6b and 6c are taken from the Columbia Object Image Library (COIL-100).*

level by a list of effects caused by the acquisition chain components on the signal (*e.g.*, noise, geometric distortion, illumination defect) and not by a list of that very components (*e.g.*, camera, lens). For example, knowing that the sensor is a "CDD camera" is not directly usable for processing images. Conversely, knowing that the CDD camera produces Gaussian noisy images is directly usable. Besides, "CDD camera" is too vague, since effects differ from camera to camera and evolve in time. Thus, 'noise' should be retained as an image processing concept, whereas 'sensor type' should not.

2. The *perceptive level* is concerned with the visual rendering of the image content without any reference to objects of interest. A definition at the perceptive level corresponds to a "syntactic" description made from a description of visual primitives, such as region, edge, background, and point of interest.

3. The *semantic level* is focused on the objects of interest. The notion of object of interest is understood from a phenomenological standpoint, namely based on the visual appearance. Hence, an object of interest

does not necessarily correspond to a scene object, but only to a part of a scene object or, on the contrary, to an aggregate of several scene objects. And a scene object can be represented by several objects of interest. The semantics of the scene is expressed thanks to information that has to be considered as relevant for discriminating one object of interest from one another. This information refers either to the individual visual description of the objects or to the description of their spatial relationships. It means that objects are identifiable by their intrinsic characteristics or by their spatial relations.

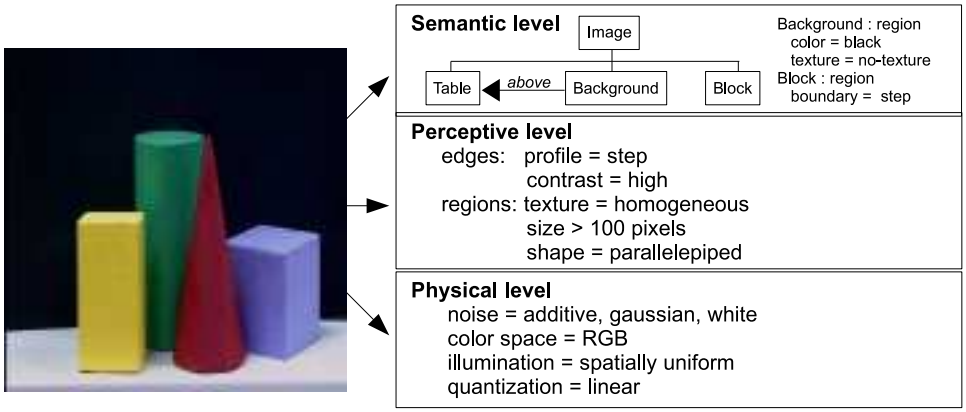


Figure 7: An image is analyzed at three levels: (1) physical, (2) perceptive, and (3) semantic.

These three levels cover both the sensory and semantic gaps since the physical level aims at bridging the sensory gap, and the perceptive and semantic levels the semantic gap.

3.1.3 Conceptual Model of the Image Class Representation

The resulting conceptual model is shown on Fig. 8. Image classes are analyzed according to the three levels: physical, perceptive, and semantic. Depending on the available knowledge about the problem, the definition of the image class at each level may be more or less important. In particular, the perceptive level is only described in the absence of information at the semantic level since a semantic description conveys more information than a perceptive description. For example, let us consider three distinct applications:

- In the case of our aerial imagery application, a large amount of information is described at the physical and semantic levels, since the

acquisition system and the scene are well known and the objects are predictable and describable. Thus, description at the perceptive level is needless.

- In case of a content-based image retrieval application, very little information about the effects of the image acquisition can be known in advance and the objects are unpredictable. The definition of the image class is thus limited to a description at the perceptive level, for example in terms of region shape or texture [7], and the physical level, often simplified to the presence of an additive white Gaussian noise.

- In case of a robotics application, engineers have a good mastery of the image acquisition process but objects may be considered as unpredictable because they usually come in large numbers and are of widely varying appearances. Thus, this kind of application will conveniently be formulated at the physical level and at the perceptive level in terms of points of interest and edge-segments.

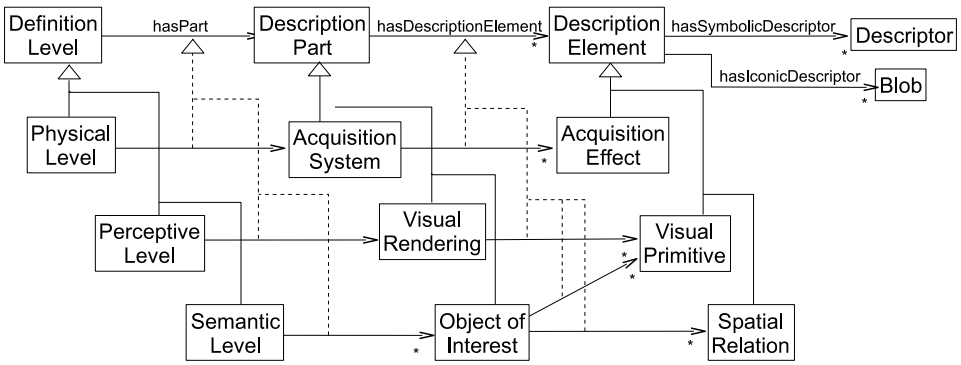


Figure 8: UML diagram of the conceptual model for defining image classes.

3.1.4 Descriptor Values

The descriptors should account for the fact that some elements of the formulation can advantageously be described by means of a linguistic vocabulary (*symbolic descriptors* with numerical and symbolic values) and other by means of an iconic vocabulary (*iconic descriptors* with image values). Among numerical values, one can choose between single values, interval of values, set of values, vector, and matrix of values (*e.g.*, a geometrical distortion matrix). Symbolic values are conventional symbolic values (*e.g.*, color space: {RGB,

HSL, YUV}), qualitative terms (*e.g.*, forest has a low brightness), and superlative terms (*e.g.*, hedge has the smallest size). Since only discriminative descriptions are considered, comparative terms (*e.g.*, has a larger size than) need not be included.

Iconic values are represented by entire images or blobs. Patches are not considered in our formulation model, because they are difficult to use manually to formulate an objective.

3.1.5 Image Variability: Notion of Context

Whatever the definition mode of the image class is, by intension or by extension, the variability of the input images sometimes makes the formulation impossible to complete. For example, let us consider the case of a daily use of the aerial imagery application. The vegetation is changing significantly during the year and therefore images acquired in spring do not have the same characteristics as images acquired in winter. To reduce such kind of variability, V. Martin et al. [30] propose to divide automatically the image class into several consistent *contexts*. For still images, contexts are automatically obtained by an unsupervised clustering algorithm of type "Density-Based Spatial clustering" [16], based on color histograms. Consequently, one has to consider as many applications as image class contexts. Applied to our aerial application, this algorithm separates the images into four contexts, which correspond to the four seasons, since landscape main colors change significantly from one season to another.

3.2 Goal Specification

Real-world image processing objectives often involve more than one single goal. To reduce the complexity of formulation, the scope of an application objective is narrowed down to a unique task. This avoids dealing with the *frame problem*. The frame problem arises when the system has to describe the effects of operations on the initial image class definition. A sequence of several tasks forces the system to propagate the effects of one task to the image class definition of the following tasks. Unfortunately, the determination of the effects of image processing operations on images is known to be an extremely difficult task. To avoid the frame problem, each task will be considered separately.

3.2.1 A Specification Combining Task and Examples

The goal is specified by a task with a related network of constraints and

optional reference images. The task is used for its ability to account for precise requirements and reference images are used for their ability to anchor the task in the data. Four types of constraints can be associated to a task:

- The *criteria to be optimized* identify the elements on which the task should focus. Examples of criteria are "maximize boundary location" and "maximize detection hits".
- The *levels of detail* determine upper and lower bounds of the task in order to refine its scope. Examples of detail levels are "separate just-touching objects" and "do not affect region shape".
- The *performance criteria* specify processing speed limits.
- The *quality criteria* express requirements on the ability of the application to perform the task within specified and unspecified conditions. Available ability values are 'reliability' or 'robustness'.

In order to reach compromise in case of doubt about compliance with a constraint, each criterion to be optimized and level of detail allow *acceptable errors*. For example, if there is a doubt whether two objects are overlapping or just-touching, an acceptable error may give a preference to the separation of the objects.

The reference images are used to supplement the specification of the task by providing examples of expected results. A reference image can either be a representation of the complete result (segmented image) or a sample of the result (contour or region sample).

3.2.2 Conceptual Model for the Goal Representation

The resulting conceptual model for the goal specification is shown on Fig. 9.

4 Ontology for Image Processing Objective Formulation

Ontologies have been widely used for designing and implementing high level scene interpretation [35, 46, 33], but seldom for low level processing. In this section, we describe an ontology that provides the primitives and their meaning for defining a language able to represent image processing objectives [41]. The ontology is a lattice of computational concepts organized upon the

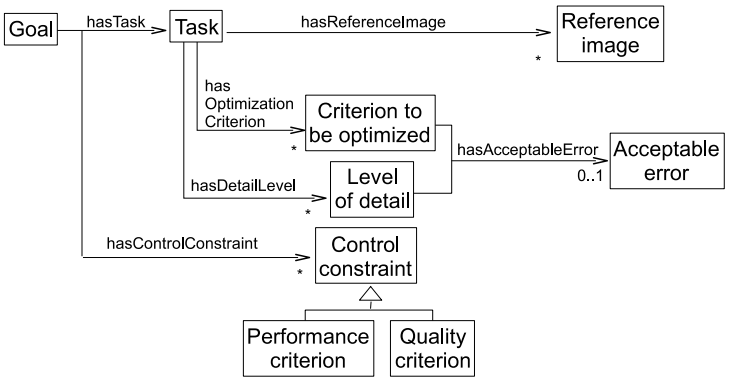


Figure 9: UML diagram of the conceptual model for specifying image processing goals.

above conceptual model. It has been operationalized with OWL DL [2] and contains 279 concepts, 42 roles, and 192 restrictions.

4.1 Physical Level Concepts

The concepts at the physical level correspond to the effects of the acquisition components on the digital image representation. Therefore, the analysis of the various components of a standard acquisition system—lighting, environment, optical system, sensor, analog-to-digital converter and storage—gives the list of their possible effects (*cf.* Fig. 10). For example, sensor optical system can generate illumination, geometry, and blur defects on images.

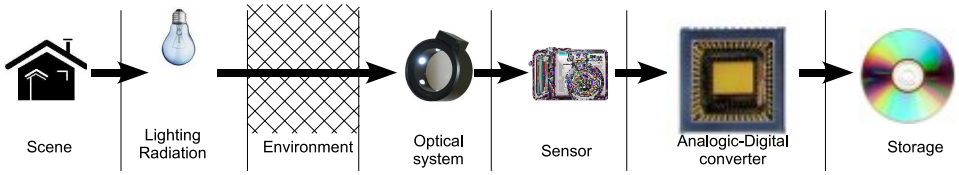


Figure 10: The analysis of the effects generated on image representation by the various components of a standard image acquisition system provides the concepts at the physical level of the ontology.

As a result, we identify nine categories of concepts (see Fig. 11): blur, noise, colorimetry, illumination, geometry, photometry, sampling, quantization, and storage. Each concept is described by a list of symbolic or iconic

[2] http://www.greyc.ensicaen.fr/~regis/Pantheon/resources/ImageProcessingOntology.owl

descriptors with the two roles *hasSymbolicDescriptor* and *hasIconicDescriptor* as detailed in Table 1. For a particular formulation, only primitives that have a genuine manifestation in the input images should be provided. Several instances of a same category can be defined, for example when different types of noise are present in the input images.

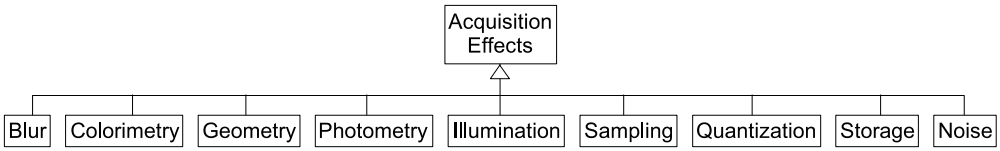


Figure 11: The concepts at the physical level correspond to the acquisition effects on input images.

4.2 Perceptive Level Concepts

The concepts at the perceptive level are the six visual primitives (see Fig. 12): region, edge, background, point of interest, image area, and cloud of points. Visual primitives are described by a list of symbolic and iconic descriptors with the two roles *hasSymbolicdescriptor* and *hasIconicdescriptor* as detailed in Table 2 and Table 3.

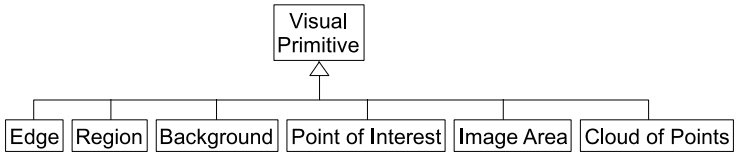


Figure 12: The concepts at the perceptive level correspond to the visual primitives.

One can notice that our representation of some concepts is quite simple. This is because we only keep descriptors that are deemed useful for choosing and evaluating image processing algorithms and not for designing them. For instance, texture features, such as wavelet or Haralick descriptors, are needless for choosing a texture-based segmentation algorithm. Only the scale (micro or macro), the type (*e.g.*, periodic, complex, dot) and orientation features are required. Other texture features will be computed directly inside the selected algorithms if need be. The same reasoning can be applied to color features since several color spaces exist. Only the HSL color space is considered for specifying the color features because it is meaningful for human beings. Other color space features may be computed inside the algorithms.

Table 1: List of the physical concepts and the descriptors linked with the two roles *hasSymbolicDescriptor* and *hasIconicDescriptor*.

Acquisition Effect	**Descriptor**
Blur	Model: *Blob, Image* Direction: *Numeric* [degree] Length: *Numeric* [pixel] Strength: *Level*[*]
Colorimetry	Model: *Blob, Image* Defect: {Bayer effect, chromatic aberration} Hue dynamics: *Numeric, Level*[*] Saturation: *Numeric, Level*[*] Colorspace: {RGB, HSL, LUV, YUV, gray, binary}
Illumination	Model: *Blob, Image* Illumination spatial: {heterogeneous, homogeneous} Illumination temporal: {stable, varying} Illumination defect: {saturation, lag, shift, blooming, smearing, flicker}
Photometry	Model: *Blob, Image* Global contrast: *Numeric, Level*[*] Dynamics: *Numeric, Level*[*] Brightness: *Numeric, Level*[*]
Quantization	Bit per pixel: *Numeric* Function: {linear, logarithmic}
Geometry	Model: *File* Defect: {astigmatism, coma, geometric distortion, spherical aberration}
Sampling	Defect: {aliasing, moiré, partial volume effect} Spatial resolution: *Dimension* [pixel
Storage	Defect: {block effect} Number of bands: *Numeric* Number of looks: *Numeric*
Noise	Model: *Blob, Image* Composition: {additive, multiplicative, mixed} Distribution: {exponential, Gaussian, uniform, Poisson, Rayleigh, impulse} Power Spectral Density: {white, pink, colored, ...} Signal Noise Ratio: *Numeric, Level*[*] Stationarity: {yes, no} First order: *Numeric, Level*[*] Second order: *Numeric, Level*[*] Third order: *Numeric, Level*[*] Fourth order: *Numeric, Level*[*]

[*] *Level: {null, very low, low, medium, high, very high}*

4.3 Semantic Level Concepts

At the semantic level, concepts describe objects of interest individually and specify spatial relationships between the objects. The set of objects is organized into a tree according to the meronymic relation "part-of". Abstract objects can be defined to factor out common description elements of objects of interest thanks to the hyperonymic relation "kind-of".

An object definition is made of the six visual primitives already identified at the perceptive level (see Table 2 and Table 3). Among the existing formal spatial relations [43], only the topological (*e.g.*, RCC-8 model [10] for 2D images) and extrinsic spatial relations (*e.g.*, on the left, on the right, above,

Table 2: List of the perceptive concepts and the descriptor categories linked with the two roles *hasSymbolicDescriptor* and *hasIconicDescriptor*.

Visual Primitive	**Descriptor Category**
Region	Model, Boundary, Photometry, Colorimetry, Texture, Region-Morphology, Orientation, Position, Region-Size, Topology
Edge	Model, Photometry, Colorimetry, Edge-Morphology, Orientation, Position, Edge-Size
Background	Model, Photometry, Colorimetry, Texture
Point of Interest	Model, InterestPoint-Morphology, Position, Photometry, Colorimetry
Image Area	Model, Photometry, Colorimetry, Texture, Region-Morphology, Orientation, Position, Region-Size
Cloud of Points	Model, Photometry, Colorimetry, Region-Morphology, Orientation, Position, Region-Size

in front of, at a distance of—where the reference frame is the image) are represented. To our knowledge, these are the only ones actually used in image processing algorithms.

4.4 Task Concepts

Tasks are specified by a verb and one optional argument. Available verbs are listed in Table 4 and cover the six categories of image processing objectives. The argument specifies the element on which the verb acts. It can be an object of interest identified at the semantic level (*e.g.*, Extract <object>), or a visual primitive defined at the perceptive level (*e.g.*, Detect <edge>) or a general image property altered by the acquisition system and characterized at the physical level (*e.g.*, Correct <noise> or Enhance <colorimetry>). Notice that Detect <object> is just the detection of the presence of the object whereas Extract <object> includes the localization of the object.

4.5 Constraint Concepts

The constraints and their related acceptable errors are represented as predefined phrases. They are listed in Table 5, Table 6, and Table 7. More particularly, there exist two possible values for the quality criterion of the control constraints:

- *"Reliability"* limits the system applicability to the conditions specified in the objective. It means that the quality of the result is the most important constraint, and thus it is better to do nothing than to yield poor results when processing unexpected input images. For example, this is a critical criterion when dealing with digitization of cultural heritage materials.

Table 3: List of the perceptive descriptor category concepts with their sub-concepts.

Descriptor Category	**Descriptor**
Model	Model: *Blob*
Edge-Morphology	Contrast: *Numeric, Level** Shape: {curve, straight line} Status: {open, close, loop} Profile: {roof, peak, step} Straightness: *Numeric, Level**
Boundary	Nature: {edge, limit of texture, limit of homogeneous regions, no boundary} Reference: **cf. the visual primitive description: edge**
Photometry	Brightness:*Level**, *Numeric, Image* Contrast: *Level**, *Numeric, Image* Model: *Patch, Image*
Colorimetry	Hue: *Numeric, Level** Saturation: *Numeric, Level** Lightness: *Numeric, Level**
Texture	Direction: *Numeric* [deg], {horizontal, vertical, oblique} Scale: {macro, micro} Type: {no-texture, contour, dot, complex, periodic}
Topology	Number of holes: *Numeric*
Orientation	Orientation: *Numeric*, {vertical, horizontal}
Position	Center of mass: *Coordinate*
Edge-Size	Length: *Numeric* [pixel],*Level** Thickness: *Numeric* [pixel],*Level**
Region-Size	Area: *Numeric*[pixel^2],*Level** Volume: *Numeric*[pixel^3],*Level** Bounding Box: *Numeric* [pixel^2|pixel^3], *Level** Diameter: *Numeric* [pixel], *Level** Thickness: *Numeric* [pixel], *Level** Net perimeter: *Numeric* [pixel], *Level** Convex perimeter: *Numeric* [pixel], *Level**
Region-Morphology	Shape: {square, rectangular, circle, ellipsoid, parallelepiped, cubic, sphere} Compactness: *Numeric*,*Level** Convexity: *Numeric*, *Level** Elongation: *Numeric*,*Level** Number of angles: *Numeric* Rectangularity: *Numeric*,*Level** Roundness: *Numeric*,*Level**
InterestPoint-Morphology	Junction type: {L, T, X, Y}

**Level: {null, very low, low, medium, high, very high}*

- *"Robustness"* extends the system applicability to unspecified conditions. It means that the production of a result is the most important constraint, and thus it is better to produce partial and even poor results than no result at all. For example, this is an important criterion for robotic vision systems.

Table 4: List of the image processing task concepts of the ontology.

Objective	**Task**
Compression	Compress
Detection	Extract <object> Detect <object> Detect <point of interest> Detect <edge>
Enhancement	Enhance <photometry> Enhance <colorimetry> Enhance <blur>
Segmentation	Partition
Restoration	Correct <noise> Correct <photometry> Correct <colorimetry> Correct <geometry> Correct <storage> Inpaint
Reconstruction	Reconstruct-shape Reconstruct-depth Reconstruct-motion

Table 5: List of the criteria to be optimized linked to the tasks by the role *hasOptimizationCriterion*, with their related acceptable errors.

Task	**Criterion to be optimized**	**Acceptable Error**
Compress	Maximize compression rate Maximize image quality	
Detect	Maximize hits	{Prefer miss to false alarm, Prefer false alarm to miss}
Enhance	Maximize fine detail visualization	
Extract	Maximize hits Maximize boundary localization	{Prefer miss to false alarm, Prefer false alarm to miss} {Prefer boundary inside, Prefer boundary outside}
Partition	Maximize segmentation precision	{Prefer sub-segmentation, Prefer over-segmentation}

5 Experimentations

Defining quantitative measures to evaluate the formulation language is a difficult task since experimental results widely depend upon the performances of the system used to automatically develop application programs from queries formulated with this language. As a validation protocol, we propose to investigate a more qualitative assessment: the expressive power of the formulation language. First, this expressive power is assessed by the ability of language to represent various formulations of a same application obtained by reverse engineering of solutions proposed in the literature. We want to show that each representation sticks to one point of view about the application and that it contains necessary information that justifies the use and the parametrization of the related solution.

Experimentations have been conducted from reverse engineering of a dozen of application domains (*e.g.*, license plate recognition, text detection, film restoration, etc.). In order to conduct experimentations, a first prototype

Table 6: List of the levels of detail linked to the tasks by the role *hasDetailLevel*, with their related acceptable errors.

Task	**Level of detail**	**Acceptable Error**
Detect	Need all pixels	{Prefer more to less, Prefer less to more}
	Need at least one pixel	
	Exclude borders touching	
Enhance	Do not affect colors rank	
	Do not affect colors ratio	
	Do not add new pixel values	
	Do not affect region shape	
	Do not affect edge profile	
Correct	Do not affect colors rank	
	Do not add new pixel values	
	Do not affect edge profile	
	Do not affect colors ratio	
	Do not affect region shape	
Extract	Put boundary inside / outside	
	Do not separate aggregate	
	Separate all	{Prefer separation, Prefer no separation}
	Separate just-touching	{Prefer separation, Prefer no separation}
	Regularize contours	
	Exclude borders touching	
	Reach sub-pixel precision localization	

Table 7: List of the control constraints linked to the tasks by the role *hasControlConstraint*.

Constraint	**Category**	**Qualifier**
Performance criterion	Optimization	{runtime-limit, real-time}
Quality criterion	Ability	{reliability, robustness, best compromise}

of a system oriented towards the production of customized programs has been implemented [41]. The program generator is Borg [9], a knowledge-based system based on a blackboard architecture that accepts requests complying with our ontology and that generates executable image processing programs. The ontology is used by the system through the OWL-API. For each application domain, the system is first configured with various suitable algorithms found in the literature. Then, the system is used with a given application ontology built by a user for a particular application from which the system selects and tunes the best algorithm.

In this section, we describe the case of the aerial imagery application presented in the introduction. We study four different solutions that have been proposed by different authors [34, 6, 15, 47]) and proposed suitable related formulations that motivate the selection and the parametrization of each solution. Second, the expressive power is assessed by the ability of the language to account for precise requirements. We demonstrate that a same task can lead to a variety of objectives simply by varying arguments, constraints and reference images.

5.1 Representation of Aerial Imagery Objectives

5.1.1 Goal Specification

The objective of localization of each potential vegetation area is divided into three sequential image processing tasks:

1. Correct storage defects: it aims at correcting compression defects;
2. Extract urban areas: it aims at removing urban areas from the image because we consider they hinder the full achievement of the next task;
3. Extract agricultural fields.

Table 8 details the specification of the *extract agricultural field* task. The related constraints are motivated by the fact that the post-processing of this task is a region classification that uses photometry, morphology, location, and size measurements. The first criterion to be optimized is the detection rate and the related acceptable error gives preference to false detections over misdetections. False detections can be eliminated during the classification step. The localization of the agricultural field boundaries is the second criterion to be optimized. In case of doubt about the location of a boundary, the related acceptable error gives preference to processing techniques that put boundary inside the region, in order to avoid photometry measurement errors with pixels outside of the field. The level of detail indicates that it is necessary to separate just-touching fields. In cases where it is difficult to decide whether a region corresponds to one or more fields, the acceptable error indicates a preference to separation, which means that over-segmentation is preferred over under-segmentation. For internal consistency reasons, it is a better solution to have several regions for a same field rather than several fields in a same region.

finally, because reproducibility is the key point of this application, robustness is preferred to reliability.

Table 8: The representation of the goal 'extract agricultural field'.

Criterion to be optimized (*acceptable error*)	Maximize hits (*prefer false alarm to miss*) Maximize boundary localization (*prefer boundary inside the region*)
Level of detail (*acceptable error*)	Separate just-touching (*prefer separation*)
Performance criterion	Optimization = runtime-limit < 30s
Quality criterion	Ability = robustness

5.1.2 Physical Definition for the Correct Storage Goal

The physical level definition of the image class for the correct storage goal is presented in Table 9. In this definition, it is noticeable that the image storage produces a block effect due to the JPEG compression. The transfer function is defined as linear. The noise is described as white Gaussian with a zero mean and a low standard deviation. This definition leads to choose a smoothing algorithm that reduces the block effect. So, the two next goals use the same physical definition but without the storage defect.

Table 9: Description of the aerial imagery application at the physical level for the correct storage task.

Acquisition effect	**hasDescriptor**
Colorimetry	Color space = RGB
Quantization	Function = linear
Noise	Composition = additive Distribution = Gaussian Power Spectral Density = white noise Mean = 0 Standard-deviation = low
Storage	Defect = block effect

5.1.3 Semantic Definition for the Extract Field Task

Since all objects of interest are predictable and describable, the definition of the image class is done at the semantic level. The tree of interest objects is given in Fig 13. In this tree, a landscape scene is considered as composed of juxtaposed objects, specified by the 'externally connected' topological relation between the geographic objects.

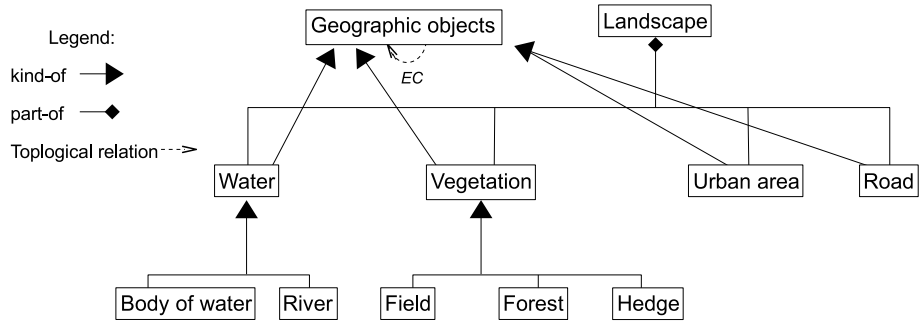


Figure 13: A landscape scene is composed of juxtaposed body of water, vegetation, urban area, and road objects. The juxtaposition of the geographic objects is represented by the topological relation "externally connected" (EC).

Three different solutions proposed in the literature [34, 6, 15, 47] are studied for performing the extract field task. Each one is motivated by a point of view about the semantic definition of the agricultural field object.

(1) The first approach is based on a region growing method controlled by edges. For example, M. Butenuth et al. [4] propose to use a watershed segmentation controlled by the gradient image, where markers are imposed as minima of the gradient image. They justify the use of this algorithm by the fact that the field areas are homogeneous regions (*Texture.Type: no texture*) and the field boundaries are highly contrasted (*Edge-Morphology.Contrast: high*). M. Mueller et al. [34] propose to control the region growing process with the edges extracted from a prior edge detection algorithm. They consider that field boundaries are well defined, with high brightness contrast to neighboring regions (*Edge-Morphology.Contrast: high*) and a typically long and straight shape (*Edge-Morphology.Shape: straight line; Edge-Size.length* $>$ *low*). In both solutions, a second step is then carried out to merge neighboring regions with low gray level difference (*Texture.Type: no texture*) and small regions into larger ones (*Region-Size.Area: level* $>=$ *low*). A final step is performed in order to fulfill the 'maximize boundary location' constraint. M. Butenuth et al. use the snake algorithm. The snakes are initialized as the smallest bounding rectangle of each extracted field since fields are parallelepiped regions (*Region-Morphology.shape: parallelepiped*). The external energy is the absolute value of the gradient (*Edge-Morphology.Contrast: high*) and the internal energy is set so as to favor rigidity because the field boundaries are described as straight and long edges (*Edge-Morphology.Shape: straight line; Edge-Size.Length* $>$ *low*).

Table 10 represents an excerpt of a semantic definition that leads the system to select and parametrize algorithms implementing the region growing approach.

Table 10: A semantic definition that supports the region growing approach.

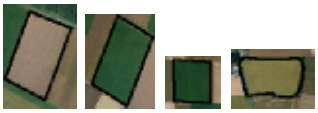
Object of Interest	**hasDefinitionElement**	**hasDescriptor**
Field	Region (field area)	Texture.Type = no texture Texture.Scale = micro Texture.Direction = 0 degree Colorimetry.Hue = $[\pi/6,\pi/2] \cup [2\pi/3,4\pi/5]$ Region-Morphology.Shape = parallelepiped Region-Size.Area >= low Boundary.Nature = edge
	Edge (Field Boundary)	Edge-Morphology.Contrast = high Edge-Morphology.Shape = straight line Edge-Size.Length > low

(2) The second approach is based on variational methods. G. Cao et al. [6] use active curve evolution via a level set algorithm to partition the

input images. The level sets are parametrized with texture features (modeled by a 3-level wavelet decomposition). The authors assume that region classes can be discriminated by a complex rotation invariant micro texture (*Texture.Type: complex; Texture.Scale: micro; Texture.Direction: 0*). The algorithm uses samples of each class of agricultural field to automatically compute texture features (*Model: blobs*).

Therefore, the image class definition given in Table 11 can be used to select and control level set algorithms.

Table 11: A semantic definition that supports the variational approach.

Object of Interest	**hasDefinitionElement**	**hasDescriptor**
Field	Region (field area)	Texture.Type = complex Texture.Direction = 0 degree Texture.Scale = micro Model =  ...

(3) The third approach is based on supervised pixel classification. This approach needs a training step from sample images. M-P. Dubuisson-Jolly et al. [15] use a maximum likelihood classification that combines texture and color information. The RGB color and micro-texture features are computed with a simultaneous autoregressive (SAR) model, from the blobs (*Texture.Type: complex; Texture.Scale: micro; Colorimetry.Hue: [π/6,π/2] $\cup$ [2π/3,4π/5]; Model: blobs*). F. Xu et al. [47] propose to use SVM for pixel classification. The SVM is based on seventeen micro texture features whose values are extracted from the blobs (*Texture.Type: complex; Texture.Scale: micro; Model: blobs*).

The selection of these two statistical approaches is motivated by the semantic definition given in Table 12, where fields are supposed to be discriminated from other regions by texture only, or by conjoint use of texture and color.

Table 12: A semantic definition that supports the statistical approach.

Object of Interest	**hasDefinitionElement**	**hasDescriptor**
Field	Region (field area)	Texture.Type = complex Texture.Scale = micro Texture.Direction = 0 degree Colorimetry.Hue = [π/6,π/2] $\cup$ [2π/3,4π/5] Model =  ...

5.2 Specification of Precise Objectives

The second example illustrates that combining the specification by task and by example enables a wide variety of nuances as well as a precision in the specification of objectives. The same basis image enhancement task can be derived into several different objectives:

1. The task 'enhance <photometry>' covers over-exposed, under-exposed or low contrasted image enhancement. The task 'enhance <colorimetry>' aims at enhancing too washed out or too saturated images and the task 'enhance <blur>' corresponds to image sharpening.

2. The associated constraint can then be used to refine the task scope. For example, the task '<enhance photometry>' with the level of detail 'do not affect color rank' restricts enhancement to global contrast and brightness modifications by lut transforms, whereas with 'maximize fine detail visualization' as criterion to be optimized the task allows more radical modifications such as histogram equalization or homomorphic filtering.

3. Finally, adding reference images to the task 'enhance <photometry>' means that the photometry enhancement should be based on the model specified by the reference images. This can be achieved by histogram matching.

6 Conclusion

In this paper, a computational model and a corresponding domain ontology for the representation of image processing application objectives have been proposed. Compared to other work that uses ontologies to capture a priori information about image analysis applications (*e.g.*, [33, 22, 46, 35, 29]), our contribution is original in the sense that only information at the image processing level is considered and all image processing objectives are covered. We bet on the existence of an image processing domain with its own concepts and vocabulary, which is independent from application domains and from which it is possible to represent the objective of any application. This is achieved by building the model and the ontology upon the three following assumptions:

1. An application is defined for one goal and one context in one image class. The intent is to limit the input image variability.

2. The goal is stated from a task with constraints and optional reference images. The specification by task enables precise identification of the goal and the taking into account of the user requirements. The reference images link the task to the image reality.

3. The definition of the image class relies on the phenomenological hypothesis and is based on a three-level denotation of relevant information: physical, perceptive, and semantic denotations. Depending on whether pieces of information are more accurately provided by examples or by a linguistic description, an extensional or an intensional denotation of the relevant information is used.

Several experiments have been carried out by reverse engineering of existing applications (such as the aerial imagery application) and by engineering of new applications to assess the expressive power of the formulation language. These experiments allow us to conclude that, although expressing the semantics by means of purely syntactic representation is not sufficient to capture the actual meaning of an image [44], it is nonetheless sufficient to design image processing programs. In the context defined by the syntactic semantics, semantic understanding is considered as the process of understanding an application domain in terms of the image processing domain [39]. The meaning of an application of a given domain is represented as a set of relationships between purely syntactic or iconic structures of the image processing domain.

The benefit of such formulation language is obviously to help image processing developers make the production of applications more robust and more reliable. An objective representation is considered as a contractual document, which reflects a consensus between the application domain specialist who sets the problem and the developer who produces the application. The explicit representation of objectives is also a means for the reuse and even the mere reproduction of existing solutions. Because of the lack of such formalization, solution reuse is too seldom exploited in image processing [48].

This work opens new perspectives for the design of automatic software generation systems. We are currently developing a human-computer interaction system which collaborates with users to compose programs by co-construction between what the users expect and what the system is able to produce. Since extensional definitions of the image class are easy to provide for users and the system uses intensional definitions for problem solving, the system performs several formulation cycles to gradually build an intensional definition from an initial extensional definition, by using interactive feature extraction and feature selection with the generation of intermediate results from test images.

We also propose to reconsider our software generator Borg, so as to avoid the pitfall of the knowledge acquisition process. This new version of the software generator will be based on the case-based reasoning paradigm. This paradigm has already been used in image processing [36, 18] and in particular by our team [17]. A case is an image processing solution in terms of a chain of algorithms. We propose to store the case with the formulation for which it has been developed. Then, the generation of a new application consists in retrieving and revising a 'similar' case, from the analysis of the formulation contents.

Finally, we also plan to use this ontology to automatically generate evaluation rules. The formulation provides an explicit representation of the purpose and the semantics of the application from which it is possible to draw up customized discrepancy metrics between a result and a reference. Such metrics can then be used by systems that use learning techniques to select the best algorithm [30]. The system is trained on test images to choose the algorithms and their parameter values yielding the results closest to the references.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, dec 2004.

[2] T. Berk, L. Brownston, and A. Kaufman. A New Color-Naming System for Graphics Languages. *IEEE Computer Graphics and Applications*, 2(3):37–44, 1982.

[3] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, and M.G. Strintzis. Semantic Annotation of Images and Videos for Multimedia Analysis. In *Second European Semantic Web Conference (ESWC)*, pages 592–607, Crete, Greece, may 2005.

[4] M. Butenuth, B-M. Straub, and C. Heipke. Automatic Extraction of Field Boundaries from Aerial Imagery. In *KDNet Symposium on Knowledge-Based Services for the Public Sector*, pages 14–25, Bonn, Germany, 2004.

[5] G. Cãmara, M. Engenhofer, F. Fonseca, and A.M. Monteiro. What's In An Image? In *Int. Conf. on Spatial Information Theory: Foundations of Geographic Information Science*, volume 2205, pages 474–488, Morro Bay, CA, sep 2001.

[6] G. Cao, Z. Mao, X. Yang, and D. Xia. Optical Aerial Image Partitioning Using Level Sets Based on Modified Chan-Vese Model. *Pattern Recognition Letters*, 29(4):457–464, 2008.

[7] M. Ceccarelli, F. Musacchia, and A. Petrosino. Content-based Image Retrieval by a Fuzzy Scale-space Approach. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 20(6):849–868, 2006.

[8] S.A. Chien and H.B. Mortensen. Automating Image Processing for Scientific Data Analysis of a Large Image Database. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8):854–859, aug 1996.

[9] R. Clouard, C. Porquet, A. Elmoataz, and M. Revenu. Borg: A Knowledge-Based System for Automatic Generation of Image Processing Programs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(2):128–144, feb 1999.

[10] A.G. Cohn, B. Bennett, J.M. Gooday, and N. Gotts. RCC: A calculus for Region based Qualitative Spatial Reasoning. *GeoInformatica*, 1(1):275–316, 1997.

[11] S. Coradeschi and A. Saffiotti. An Introduction to the Anchoring Problem. *Robotics and Autonomous Systems*, 43(2-3):85–96, 2003.

[12] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V-K. Papastathis, and M.G. Strintzis. Knowledge-Assisted Semantic Video Object Detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 15(10):1210–1224, 2005.

[13] B.A. Draper, J. Bins, and K. Baek. ADORE: Adaptive Object Recognition. In *Int. Conf. on Vision Systems*, pages 522–537, Las Palmas de Gran Canaria, Spain, 1999.

[14] B.A. Draper, A.R Hanson, and E.M. Riseman. Knowledge-Directed Vision: Control, Learning, and Integration. *Proceeding of the IEEE*, 84(11):1625–1637, 1996.

[15] M-P. Dubuisson-Jolly and A. Gupta. Color and texture fusion: application to aerial image segmentation and GIS updating. *Image and Vision Computing*, 18(10):823–832, 2000.

[16] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, 1996.

[17] V. Ficet-Cauchard, C. Porquet, and M. Revenu. CBR for the management and reuse of image-processing expertise: a conversational system. *Engineering Applications of Artificial Intelligence*, 12(6):733–747, 1999.

[18] M. Frucci, P. Perner, and G. Sanniti di Baja. Case-Based-Reasoning for Image Segmentation. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 22(5):829–842, 2008.

[19] I.B. Gurevich, O. Savetti, and Y.O. Trusova. Fundamental Concepts and Elements of Image Analysis Ontology. *Pattern Recognition and Image Analysis*, 19(4):603–611, 2009.

[20] J-I. Hasegawa, H. Kubota, and J-I. Toriwaki. Automated Construction of Image Processing Procedures by Sample-Figure Presentation. In *8th Int. Conf. on Pattern Recognition (ICPR)*, pages 586–588, Paris, France, 1986.

[21] C. Hudelot, J. Atif, and I. Bloch. Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets Systems*, 159(15):1929–1951, 2008.

[22] C. Hudelot, N. Maillot, and M. Thonnat. Symbol Grounding for Semantic Image Interpretation: From Image Data to Semantics. In *IEEE Int. Workshop on Semantic Knowledge in Computer Vision (ICCV)*, pages 1–8, Beijing, China, 2005.

[23] J. Hunter. Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In *Int. Semantic Web Working Symposium (SWWS)*, pages 261–281, Stanford, CA, 2001.

[24] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval Using Cross-Media Relevance Models. In *26th Int. Conf. on Research and Development in Information Retrieval*, pages 119–126, Toronto, Canada, jul 2003.

[25] B. Leibe and B. Schiele. Interleaved Object Categorization and Segmentation. In *British Machine Vision Conference (BMVC)*, pages 145–161, Norwich, UK, sep 2003.

[26] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transaction on Graphics*, 23(3):689–694, 2004.

[27] I. Levner, V. Bulitko, L. Lihong, G. Lee, and R. Greiner. Towards Automated Creation of Image Interpretation Systems. In *16th Australian Joint Conference on Artificial Intelligence*, pages 653–665, Perth, Australia, 2003.

[28] Q. Li, S. Luo, and S. Zhongzhi. Semantics-Based Art Image Retrieval Using Linguistic Variable. In *Fourth Int. Conf. on Fuzzy Systems and Knowledge Discovery*, pages 406–410, Haikou, China, 2007.

[29] N. Maillot and M. Thonnat. Ontology-Based Complex Object Recognition. *Image and Vision Computing*, 26(1):102–113, 2008.

[30] V. Martin. *Cognitive Vision: Supervised Learning for Image and Video Segmentation*. PhD thesis, University of Nice Sophia Antipolis, France, dec 2007.

[31] V. Martin, N. Maillot, and M. Thonnat. A Learning Approach for Adaptive Image Segmentation. In *4th IEEE Int. Conf. on Computer Vision Systems (ICVS)*, pages 40–47, New York, 2006.

[32] T. Matsuyama. Expert Systems for Image Processing: Knowledge-Based Composition of Image Analysis Processes. *Computer Vision, Graphics and Image Processing*, 48(1):22–49, oct 1989.

[33] V. Mezaris, I. Kompatsiaris, and M.G. Strintzis. Region-based image retrieval using an object ontology and relevance feedback. *Journal on Applied Signal Processing*, 6(1):886–901, 2004.

[34] M. Mueller, K. Segl, and H. Kaufmann. Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery. *Pattern Recognition*, 37(8):1619–1628, 2004.

[35] B. Neumann and R. Möller. On Scene Interpretation with Description Logics. *Image and Vision Computing*, 26(1):82–110, 2008.

[36] P. Perner, A. Holt, and M. Richter. Image Processing in Case-Based Reasoning. *The Knowledge Engineering Review*, 20(3):311–314, 2005.

[37] A. Protière and G. Sapiro. Interactive Image Segmentation via Adaptive Weighted Distances. *IEEE Trans. on Image Processing*, 16(4):1046–1057, 2007.

[38] A.R. Rao and G.L. Lohse. Towards a Texture Naming System: Identifying Relevant Dimensions of Texture. In *4th IEEE Conference of Visualization*, pages 220–227, San Jose, CA, 1993.

[39] W.J. Rapaport. What Did You Mean by That? Misunderstanding, Negotiation, and Syntactic Semantics. *Minds and Machines*, 13(3):397–427, 2003.

[40] H. Reichgelt. *Knowledge Representation: An Ai Perspective*. Ablex Publishing Corporation, Norwood, New Jersey, 1991.

[41] A. Renouf, R. Clouard, and M. Revenu. A Platform Dedicated to Knowledge Engineering for the Development of Image Processing Applications. In *Int. Conf. on Enterprise Information Systems (ICEIS)*, pages 271–276, Funchal, Portugal, jun 2007.

[42] A. Renouf, R. Clouard, and M. Revenu. How to formulate image processing applications? In *Int. Conf. on Computer Vision Systems (ICVS)*, pages 1–10, Bielefeld, Germany, 2007.

[43] G. Retz-Schmidt. Various views on spatial prepositions. *AI Magazine*, 9(1):95–105, 1988.

[44] S. Santini, A. Gupt, and R. Jain. Emergent Semantics through Interaction in Image Databases. *IEEE Trans. Knowledge and Data Engineering*, 13(3):337–351, 2001.

[45] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[46] C.P. Town. Ontological Inference for Image and Video Analysis. *Machine Vision and Applications*, 17(2):94–115, 2006.

[47] F. Xu, X. Li, and Q. Yan. Aerial Images Segmentation based on SVM. In *Int. Conf. on Machine Learning and Cybernetics*, volume 4, pages 2207–2211, Wuhan, China, nov 2003.

[48] P. Zamperoni. Plus ça va, moins ça va. *Pattern Recognition Letters*, 17(7):671–677, jun 1996.

[49] Y.J. Zhang. A Survey on Evaluation Methods for Image Segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[50] X-R. Zhou, A. Shimizu, J. Hasegawa, J. Toriwaki, T. Hara, and H. Fujita. Vision Expert System 3D-IMPRESS for Automated Construction of Three Dimensional Image Processing Procedures. In *Seventh Int. Conf. on Virtual Systems and Multimedia*, pages 527–536, Berkeley, CA, 2001.