AUTOMATIC INCREMENTAL MODEL LEARNING FOR SCENE INTERPRETATION

J. Hartz
Department of Computer Science
University of Hamburg
email: hartz@informatik.uni-hamburg.de

L. Hotz
HITeC e.V.
University of Hamburg
email: hotz@informatik.uni-hamburg.de

B. Neumann
Department of Computer Science
University of Hamburg
email: neumann@informatik.uni-hamburg.de

K. Terzić
Department of Computer Science
University of Hamburg
email: terzic@informatik.uni-hamburg.de

ABSTRACT
In this paper, we investigate automatic model learning for the interpretation of complex scenes with structured objects. We present a learning, interpretation, and evaluation cycle for processing such scenes. By including learning and interpretation in one framework, an evaluation and feedback learning is enabled that takes interpretation challenges like context and combination of diverse types of structured objectes into account. The framework is tested with the interpretation of terrestrial images of man-made structures.

KEY WORDS
Computer Vision, Image Understanding, Machine Learning, Ontologies

1 Introduction

In computer vision, growing interest in artificial cognitive systems has brought about increased efforts to extend vision systems towards capabilities for high-level vision or scene interpretation. These are terms commonly used for vision tasks going beyond single-object recognition, such as inferring the existence and location of occluded aggregate parts from already observed ones. Typical examples are monitoring tasks (e.g. detecting a bank robbery), analysing traffic situations for a driver assistance system or, as in the case of this paper, interpreting terrestrial images of complex man-made structures (e.g. facades).

As explicated in [8], scene interpretation can be formally modelled as a knowledge-based process. The burden of the interpretation process lies on the conceptual descriptions, and the richer a domain, the more demanding is the task of designing these descriptions. It is foreseeable that designing knowledge bases for larger applications using a handcrafting approach will be prohibitively error-prone.

We believe that in the long run high-level vision can only be achieved by leading the system through a supervised learning phase where the concepts for a particular domain are acquired based on examples. By employing learning methods, we can extend this approach to learn during the interpretation. In the proposed scenario, we start the interpretation with an empty knowledge base. For each instance from an annotated image that is not interpreted correctly, feedback learning steps are triggered to correct the faulty conceptual description - we *evaluate* interpretation results for improving learnt concepts.

In this paper, we present an evaluation framework for the SCENIC system [5]. The SCENIC system consists of a *high-level* layer that includes a domain-specific knowledge base of concepts (which are learnt) and an interpretation process, which propagates constraints, instantiates concepts to *instances*, determines relations between instances, etc. Concepts represent aggregate models, instances represent *aggregate instantiations* (or simply *aggregates*), i.e. ensembles of concrete objects in scenes. The interpretation process attempts to create assertations about the scene that describe the observed evidence.

A *middle layer* (or "the Matchbox") matches the instances of high-level concepts to the evidence provided by the low-level processes. It works both bottom-up (creating high-level objects from unambiguous evidence) and top-down (when attempting to confirm or refute high-level hypotheses by matching them to ambiguous evidence).

Diverse *low-level* detectors (*image processing modules, IPMs*) provide a range of responses after analysing the underlying image. Their detections are stored in the Matchbox as classified detections, or as regions described by rich feature vectors which can be matched to objects in the high-level ontology.

The evaluation framework as described in Section 2 allows for the evaluation of learnt concepts using a database of annotated images[1]. In Section 3, we describe the use of the framework for evaluating learnt concepts. Section 4 presents experiments made with terrestrial images. We conclude with a discussion of related work and a summary.

2 Evaluation Framework

Overview The idea behind the evaluation framework is to compare the interpretation result to a standard, correct

[1] www.ipb.uni-bonn.de/projects/etrims_db

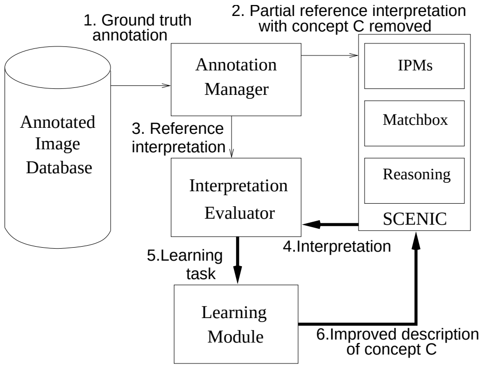

Figure 1. Modules of the SCENIC evaluation framework. Steps 4 to 6 are automatically repeated until the image is interpreted correctly. Then, a next image is provided and the process starts again from Step 1. The Evaluation Manager (not shown) controls these iterations.

interpretation. This is comparable to a teacher giving the correct interpretation and evaluating the student's performance. In our case, the teacher's input is contained in the annotated image database. The annotations comprise all instances of all known object aggregates and primitive object classes present in the scene and include the partonomical relations between them. These annotations are compared to automatically created interpretations.

The complete evaluation framework (see Figure 1) consists of the an *Annotation Manager* module, which can create a reference interpretation from the annotated scene, an *Evaluation Manager* module, which oversees the process, an *Interpretation Evaluator* module, which compares two interpretations, the *SCENIC interpretation system* (the integrated low-level image processing modules, the middle layer, the high-level interpretation module), and a *Learning* module, which is based on Version Space Learning (VSL).

As we are interested in evaluating the interpretation facilities of the learnt aggregate models, it is important that the evaluation process stays as close as possible to the standard interpretation process. By comparing the interpretation result to the one provided by the teacher, it is possible to evaluate how well a certain version of an aggregate model can be used to interpret scenes in our domain.

Annotation Manager The Annotation Manager module prepares the ground truth data file needed to perform the evaluation. A reference interpretation for the annotated scene is generated in terms of a partonomical hierarchy of scene objects and aggregates. This reference interpretation is passed to the Interpretation Evaluator module, which will compare it to the interpretation result later. A partial reference interpretation is used as input to the interpretation process. It is obtained by stripping all entities that should come out of the interpretation process. The Evaluator module later determines whether the stripped entities were correctly recognised.

Evaluation Manager As our evaluation is closely coupled with the interpretation process, no additional logic is required to perform the evaluation. The Evaluation Manager is a simple program which controls the evaluation process. In the most general case, it

- initialises all parts of the system,
- loads up a partial interpretation into the high-level system,
- starts the interpretation process, and
- compares the resulting interpretation with the intended interpretation from the database by passing them to an external evaluator.

The second step allows the evaluation to start at any point in the interpretation process, if specific scenarios need to be evaluated (an example is the evaluation of individual concepts, described later in this paper). If this is not desired, an empty interpretation is loaded, and the interpretation starts from the beginning.

Matchbox and High-Level Interpretation For evaluating learnt concept models, the whole SCENIC system is operated in its standard mode for scene interpretation. The interpretation is based on the configuration system KONWERK and uses an ontology consisting of:

- a concept hierarchy, which provides object classes (concepts) in a taxonomical and compositional hierarchy,
- constraints that represent restrictions between object classes like spatial relations or numeric restrictions,
- a task description, which specifies an aggregate and possibly additional parts and restrictions among them that have to be in the final interpretation, and
- procedural knowledge, which specifies the interpretation strategies in a declarative manner.

The main task of the interpretation is to find a logical model for the set of observed scene objects, i.e. to integrate all scene objects into aggregates corresponding to the knowledge base. The interpretation process can hypothesise scene objects if their existence is likely, considering the current interpretation context. All hypotheses made by the interpretation process should be confirmed by the evidence in the scene. In the evaluation setting described in this paper, evidence is provided by the annotations and not by IPMs (to limit any negative influence of imperfect IPMs). For confirmation, a request is sent to the Matchbox (middle layer module), which controls the image processing modules in the following way:

- The Matchbox knows the partonomical hierarchy and potential overlap of objects, to be able to allow or refute overlapping objects.
- The Matchbox receives the same filtered annotation as the high-level system as evidence.
- A confirmation request from the high-level system can have several different outcomes:
  - *confirm*, if the hypothesis fits to the evidence

- *refute*, if the hypothesis overlaps with an object of another type
- *do not know*, if the hypothesis is at an empty place or can overlap with existing objects

Interpretation Evaluator After the interpretation process has been conducted, the high-level system delivers an interpretation result which is compared to the annotated ground-truth image. All instances which play a role in the evaluation of a component are compared to instances of the same type in the annotation. The Interpretation Evaluator matches all instances in the interpretation result to those in the annotation. In this manner, true positives, false positives and false negatives can be detected. These results are gathered to derive an empirical evaluation.

```
Size and configuration

Aggregate Width =    ]549..INF] cm
Aggregate Height =   [0..200[ cm
Parts Width =        [0..INF] cm
Parts Height =       [0..INF] cm
Composition

Has-Parts =          [3..INF]
        window =     [3..INF]
        door =       [0..0]
Part-Of =            [1..1]
        facade =     [0..1]
        roof =       [0..1]

Symbolic attributes

Shape =              { Elongated-X }

Internal spatial relations

(window000) LeftNeighbourOf [132..324] cm (window001)
(window000) LeftOf [339..649] cm (window002)
(window001) LeftNeighbourOf [206..325] cm (window002)
(window001) RightNeighbourOf [132..324] cm (window000)
(window002) RightNeighbourOf [206..325] cm (window001)
(window002) RightOf [339..649] cm (window000)

External spatial relations

(concept013) BelowOf [44..1865] cm (sky020)
(sky020) AboveOf [44..1865] cm (concept013)
```

Table 1. Learnt aggregate model "Window Array". The description language used in detailed in [3].

Learning Module As described in [3], we have developed a Version Space Learning (VSL) method which generates a set of possible concept hypotheses for positive and negative examples of a given aggregate (see also [7]). A *concept hypothesis* represents a possible description of real-world aggregates presented as learning examples. By increasing the example set a concept hypothesis might change. In VSL, the space of possible concept hypotheses VS is implicitly represented through an upper and a lower bound on their generality. The General Boundary GB contains all maximally general members of VS, the Specific Boundary SB contains all maximally specific members of VS. GB and SB completely determine VS, which is the set of hypotheses h being $more\text{-}general\text{-}or\text{-}equal$ to an element of SB and $more\text{-}specific\text{-}or\text{-}equal$ to an element of GB. Roughly speaking, the unique hypothesis $h_s \in SB$ covers all positive examples and a hypothesis $h_i \in GB$ excludes all negative examples. There are multiple h_is because the negative examples can be excluded in various ways, e.g. through different spatial relations.

To select concept hypotheses from the learnt Version Space, for inclusion in our conceptual knowledge base, we choose h_s as the most specific concept hypothesis for the conceptual knowledge base and the logical conjunction of all $h_i \in GB$ as the most general concept hypothesis h_g (see also [4]). Hypothesis h_g is the most general concept hypothesis excluding negative examples by all discriminating attributes.

As an example result of the learning process, the General Boundary conjunction hypothesis h_g for the aggregate "Window Array", learnt from 13 annotated positive examples and 260 generated negative examples is presented in Table 1.

Integration of the Modules The integration of the previously discussed modules is based on the ontology used for interpretation. The learning module provides concepts and constraints of the ontology by an extension of the Web Ontology Language OWL. The annotation manager represents the reference interpretation as instances of the OWL concepts. Both, the ontology and the instances are used by the high-level interpretation and the Matchbox for constructing a complete scene interpretation. This again is represented as instances of the ontology. Thus, the ontology represents the knowledge, which is used by all modules.

3 The Concept Evaluation Procedure

In this section, we present the procedure for the evaluation of learnt conceptual aggregate models. The evaluation is based on the general SCENIC interpretation framework presented in Section 2. Based on the evaluation result, distinct feedback learning steps can be triggered.

For evaluation, we follow an *experience-gaining, incremental evaluation* approach. In this approach, the training set T^r and the test set T^s are changed incrementally by each step. Learning of concept C is done with the training set as in the usual approach. However, testing is done with one element e of the test set. After this test, the result is included in the training set and a new concept is learnt with this input. Thus, using this approach, a concept is obtained which covers the positive examples and excludes the negative examples depending on the result of the interpretation of e.

In detail: Let S be the set of images, then initially T_0^r = e with $e \in S$, T_0^s = $S \setminus e$. In a general situation, we have two sets T_n^r and T_n^s, where T_n^r is the current training set and T_n^s is the current test set. At each step, an element $t \in T_n^s$ is selected and interpreted using C. After this step, the new training set is $T_{n+1}^r = T_n^r \cup t$ and the new test set is T_{n+1}^s = $T_n^s \setminus t$. Thus, the test image becomes a new training image for improving C which leads to concept C'. Depending on the interpretation result (i.e. if instances of C could be created during interpretation or not) t is used appropriately in the succeeding learning step. Thus, the Specific Boundary or the General Boundary of C is improved. This process is performed on all images in S.

Matchbox and High-level Interpretation for VSL Evaluation For evaluating the appropriateness of C as described in Section 2, the interpretation process is performed as in the typical, i.e. non-evaluating, interpretation situation. Thus, the concept hierarchy and the constraints represent known knowledge about the façade domain including C with its learnt constraints. The task description is a given scene description derived from annotation, where all instances of C have been *removed*. The procedural knowledge describes the usual combined bottom-up and top-down interpretation strategies.

Since the annotation has been stripped of instances of concept C before the task description was generated, annotated parts p_C of C do not belong to an aggregate. The goal of the interpretation process is to check, whether these parts fulfill the concept description of C including constraints (e.g. spatial relations like `stairs below door`). In Figure 2 an example is given for a task description, i.e. a set of several views that depict primitives annotated in an image. In this case, the aggregate instance for the concept `entrance` (i.e. concept C) is not given as a view, thus, its primitives do not belong to any aggregate instance.

During the interpretation process, instances given to the high-level system through the task description are easily confirmed by the Matchbox. The Matchbox simply compares views from high-level with the ground truth data from the annotated image, which leads to confirmation.

An instance I of concept C is created, if the parts in the task description fulfill the constraints of C given in the knowledge base. If they do not fulfill these constraints, no aggregate of concept C is constructed. In detail, the different outcomes of the interpretation process are:

All parts p_C are included in I. An interpretation with an instance of C is created, where the parts p_C are included and the restrictions are fulfilled, i.e. the concept description can be accepted.

Some of the parts p_C are included in I. An interpretation is created where the parts p_C are partly included in I. Missing parts are hypothesized and a valid interpretation is created. However, the remaining parts are not included in I, which is identified by the evaluator.

All parts p_C are included in another aggregate. An interpretation with an instance of another concept is created, where the parts are included. For example, the parts are included in the `façade` aggregate, which can contain all parts. The interpretation process returns the interpretation without an instance of C for the given parts.

Concept Evaluation and Feedback Learning The resulting interpretation is returned to the Evaluation Manager. Figure 3 shows an example for a resulting interpretation. Besides aggregate instances of concept C, instances of other aggregate models (concepts) may also be constructed (here of `balcony`). These are also considered by the evaluator, since they might interfere with the expected evaluation result. If for example, an instance of `balcony` integrates a part that belongs to an instance of concept C, that aggregate cannot be recognised any more. In such a case, feedback learning would lead to a specialisation of `balcony` to prevent it from integrating that part. In a next iteration of the evaluation cycle on the same image, the proper interpretation result will then be found.

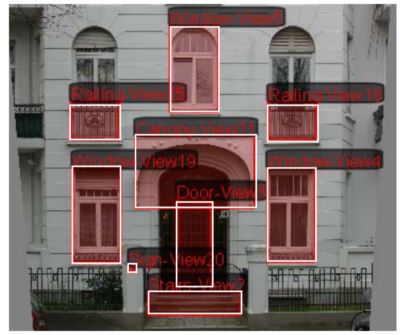

Figure 2. Primitive scene objects provided by annotation. For the visualisation here, the image has been cropped and the annotation has been stripped.

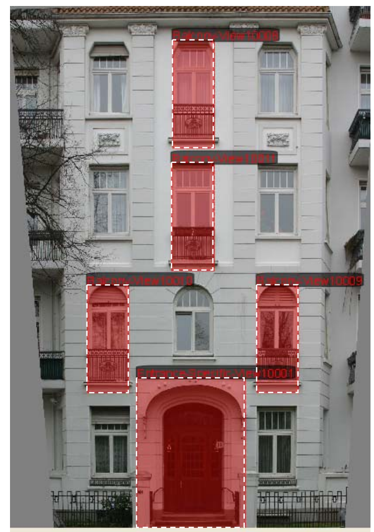

Figure 3. Constructed aggregates of concept `entrance` and `balcony` through interpretation of primitives in 2.

There can be several reasons for misinterpretation, which lead to feedback learning steps. Generally, in the case an instance is not found, the learnt concept description is too specific. As a feedback step, the unrecognised aggregate instance is introduced to the learning process as a positive example, generalizing the learnt concept description. In the case something is wrongly considered to be an instance, the learnt concept description is too general. Therefore the set of misinterpreted objects is introduced to the learning module as a negative example. For both cases another misclassification of the particular instance becomes

Figure 4. Training image

impossible, regardless which concept hypothesis is chosen from VS after feedback learning.

The cases of misinterpretation of concept C by an Instance I composed of scene objects $p_{1..n}$ can be further differentiated. We distinguish between the cases where h_s needs to be generalized, and where h_g needs to be specialised.

False negative recognitions: The instance I is not recognised by h_s or h_g or the instance I is not recognised by h_s, only by h_g. For these cases, instance I needs to be introduced as a positive example for concept C. The concept descriptions in the knowledge base are then generalized minimally to include I.

False positive recognitions: An instance I is recognised by h_g with none of the objects $p_{1..n}$ or an instance I is recognised by h_g with a subset of $p_{1..n}$ or an instance I is recognised by h_g with a subset of $p_{1..n}$ and additional objects or an instance I is recognised by h_g with all $p_{1..n}$ and additional objects. For these cases, the set of objects which has been incorrectly recognised to be an instance of C is introduced as a negative example. The concept descriptions are then specialised minimally to exclude this set of objects from being recognised as an instance of C.

4 Experiments

In this section, we present two experiments of feedback learning in practice. The first experiment describes the approach with two examples, the second demonstrates it on several examples.

In the first experiment, the concept description for `entrance` has to be refined to cover a new image. The process starts out with a conceptual description learnt from the four examples in Figure 4. The learnt internal spatial relation attribute of h_s (the concept hypothesis of the Specific Boundary) is presented in Table 2.

Using the concept description in Table 2, the image in Figure 2 is introduced for evaluation. On this image, the interpretation process is able to instantiate the learnt h_g concept of `entrance` (i.e. the General Boundary conjunction hypothesis). An instantiation of the more specific h_s concept of `entrance` is not possible, because h_s

```
Internal spatial relations

Stairs0:
BelowNeighbour, Dist = [0 0] cm, Door1
Below, Dist = [200 209] cm, Canopy2
Overlap, Dist = [0 0] cm, Railing3
BelowNeighbour, Dist = [191 191] cm, Sign4
Door1:
AboveCenterNeighbour, Dist = [0 0] cm, Stairs0
BelowCenterNeighbour, Dist = [1 7] cm, Canopy2
RightNeighbour, Dist = [23 26] cm, Railing3
LeftNeighbour, Dist = [8 8] cm, Sign4
Canopy2:
Above, Dist = [200 209] cm, Stairs0
AboveNeighbour, Dist = [1 7] cm, Door1
AboveNeighbour, Dist = [100 104] cm, Railing3
Railing3:
LeftNeighbour, Dist = [23 26] cm, Door1
Overlap, Dist = [0 0] cm, Stairs0
Below, Dist = [100 104] cm, Canopy2
Sign4:
AboveCenter, Dist = [191 191] cm, Stairs0
RightNeighbour, Dist = [8 8] cm, Door1
```

Table 2. `entrance` concept before feedback learning.

```
Internal spatial relations

Stairs0:
BelowNeighbour, Dist = [0 0] cm, Door1
BelowNeighbour, Dist = [56 191] cm, Sign4
Below, Dist = [197 209] cm, Canopy2
Overlap, Dist = [0 0] cm (MaxDist = 0), Railing3
Door1:
AboveCenterNeighbour, Dist = [0 0] cm, Stairs0
BelowCenterNeighbour, Dist = [1 7] cm, Canopy2
RightNeighbour, Dist = [26 26] cm, Railing3
Neighbour, Dist = [8 98] cm, Sign4 (*)
Canopy2:
Above, Dist = [197 209] cm, Stairs0
AboveNeighbour, Dist = [1 7] cm, Door1
AboveNeighbour, Dist = [104 104] cm, Railing3
AboveRight, Dist = [136 136] cm, Sign4 (*)
Railing3:
LeftNeighbour, Dist = [26 26] cm, Door1
Overlap, Dist = [0 0] cm, Part Stairs0
Below, Dist = [104 104] cm, Canopy2
Sign4:
Above, Dist = [56 191] cm, Stairs0
Neighbour, Dist = [8 98] cm, Door1 (*)
BelowLeft, Dist = [136 136] cm, Canopy2
```

Table 3. `entrance` concept after feedback learning. Several relations are generalized (see (*)).

is too restrictive. This is due to the spatial relations between the door and the sign, which model the sign to be `RightNeighbour` of the door (Table 2). Following the procedure described in Section 3, the interpreted instance is now introduced automatically as a positive example for `entrance`, to generalize the most specific representation.

The result of this feedback learning step is shown in Table 3. The spatial relation attribute has been generalized over the instance that could not be covered before. In the new model of the spatial structure of the aggregate parts, the sign is localised as the `Neighbour` of the door, which does not enforce a specific direction. Now the interpretation process is able to instantiate the learnt h_s concept of `entrance` with the correct parts. Further feedback learning steps with this example image are not necessary.

In the second experiment, the concept description for `balcony` has to be refined to cover several new images. The process starts out with a conceptual description learnt by one image and proceeds through several images, each of which with a number of balconies. As expected in the

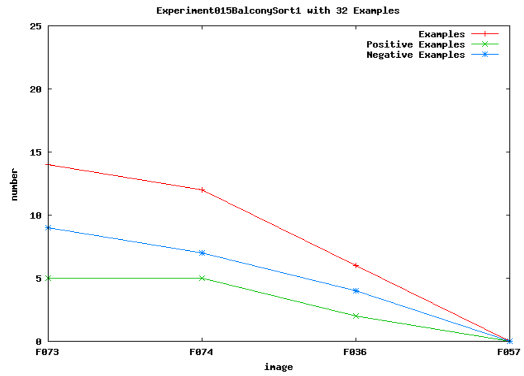

Figure 5. Reducing needed positive and negative examples when proceeding with feedback learning.

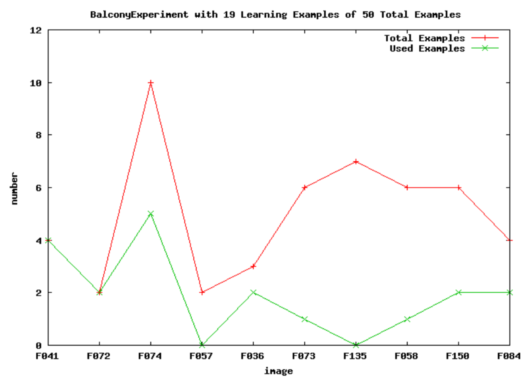

Figure 6. Number of examples per image for learning a `balcony` concept. The upper line depicts the number of balconies in each image, the lower line the number of used balcony examples for learning.

beginning of the process, more negative increments occur because the concept `balcony` is learnt on the basis of a single image. As the system learns from new images, fewer examples are needed. Balcony examples of images $F057$ and $F135$ could all be interpreted with the previously learnt concept, i.e. no further learning was needed to interpret these images (see Figure 6). Figure 5 shows an example with the behavior for positive and negative examples.

5 Discussion and Summary

In this paper, we introduce an automatic way of incremental model learning by using annotated examples. The models are used for interpreting scenes of terrestrial images and thus, we incrementally improve the interpretation facilities of the learnt models. This is done by including high-level knowledge-based interpretation techniques, low-level image processing methods, learning methods, and evaluation methods in one single framework. By coupling interpretation with learning, a feedback learning is enabled, which takes the context provided by the interpretation into account. This framework is used for evaluating the learnt models, thus, they are tested in the targeted application scenario of scene interpretation. Because an evaluation *framework* is provided, we are able to evaluate each individual part of the system.

Other knowledge-based approaches for scene interpretation are described in e.g. [6, 9]. The main difference of our approach is the incorporation of learning methods for constructing the knowledge base. This enables us to apply the approach to specific technical domains (e.g. like monitoring tasks on airports) where the construction of knowledge bases might be time consuming and general knowledge sources like WordNet [1] cannot be applied.

In [2] a similar approach is described for knowledge-based image understanding. An evaluation is also provided, but it is processed by end-users in a handcrafted way. In our approach, annotations are given by users and the evaluation is automatically done by applying membership criteria for aggregates. Furthermore, the evaluation result is used for relearning, i.e. feedback learning.

In future work, we will try to identify criteria, which make images to good learning examples and reduce the learning steps needed for obtaining the desired performance. Currently, we expect that images with aggregates which have a large number of parts are good candidates for fast learning.

Acknowledgement

This research has been supported by the European Community under the grant IST 027113, eTRIMS - eTraining for Interpreting Images of Man-Made Scenes.

References

[1] C. Fellbaum. English Verbs as a Semantic Net. *International Journal on Lexicography*, 3(4):265–277, 1990.

[2] B. Georis, M. Mazière, F. Brémond, and M. Thonnat. Evaluation and Knowledge Representation Formalisms to Improve Video Understanding. In *Proc. of IEEE International Conference on Computer Vision Systems ICVS06*, pages 27–27, Jan. 2006.

[3] J. Hartz and B. Neumann. Learning a Knowledge Base of Ontological Concepts for High-Level Scene Interpretation. In *International Conference on Machine Learning and Applications*, Cincinnati (Ohio, USA), December 2007.

[4] D. Haussler. Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artificial Intelligence*, 36(2):177–221, 1988.

[5] L. Hotz and B. Neumann. Scene Interpretation as a Configuration Task. *Künstliche Intelligenz*, 3:59–65, 2005.

[6] C. Liedtke, A. Blömer, and T. Gahm. Knowledge-Based Configuration of Image Segmentation Processes. *International Journal of Imaging Systems and Technology*, 2:285–295, 1990.

[7] T. Mitchell. Version Spaces: A Candidate Elimination Approach for Rule Learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 305–310, 1977.

[8] B. Neumann. A Conceptual Framework for High-Level Vision. Technical Report FBI-HH-B245/02, Fachbereich Informatik, University of Hamburg, 2002.

[9] C. Smyrniotis and K. Dutta. A knowledge-based system for recognizing man-made objects in aerial images. In *Proceedings CVPR '88 Computer Society Conference on Computer Vision and Pattern Recognition.*, pages 111–117, Jun 1988.