Feedback between Low-level and High-level Image Processing

Lothar Hotz[1], Bernd Neumann[2], Kasim Terzic[2],
Jan Sochmann[3]

(1) Hamburg Informatics Technology Center HITeC
(2) University of Hamburg
(3) University of Prague

Email: hotz@informatik.uni-hamburg.de

May 2007

Zusammenfassung

Systeme für rechnerbasierte Szeneninterpretation werden häufig als Erweiterung von Objekterkennungssystemen konzipiert, bei denen die für niedere Verarbeitungsstufen übliche bottom-up Verarbeitung auch auf die höhere Bilddeutung ausgedehnt wird. In diesem Bericht stellen wir ein generisches Szeneninterpretationssystem vor, das Hypothesen generieren und top-down Verarbeitungsschritte anstoßen kann. Dadurch ist es möglich, örtlich fokussierte und spezifisch parametrierte Bildanalysen durchzuführen. Es werden experimentelle Ergebnisse zur Erkennung von Strukturen in Gebäudefassaden vorgelegt.

Abstract

Scene interpretation systems are often conceived as extensions of low-level image analysis with bottom-up processing for high-level interpretations. In this paper we show how a generic high-level interpretation system can generate hypotheses and start top-down analysis. This allows for spatially focussed and specifically parametrised image analysis steps. Experimental results of the recognition of structures in building facades are reported.

1 Introduction

In recent years, growing interest in artificial cognitive systems has brought about increased efforts to extend the capabilities of computer vision systems towards higher-level interpretations [2, 16, 23, 19, 7, 6, 12]. Roughly, a high-level interpretation can be defined as an interpretation beyond the level of recognised objects. Typical examples are monitoring tasks (e.g. detecting a bank robbery), analysing traffic situations for a driver assistance system or interpreting aerial images of complex man-made structures. While existing approaches to high-level interpretation differ in many respects, they have in common that prior knowledge about spatial and temporal relations between several objects has to be brought to bear, be it in terms of probabilistic models [19], frame-based models called aggregates, logic-based conceptual descriptions [16], Situation Graph Trees [15] or Scenarios [5]. In the following, we will use the term "aggregate" for meaningful multiple-object units of a high-level scene interpretation.

In extending vision to high-level interpretations, one of the challenges is to exploit expectations derived from high-level structures for improved low-level processing. There exists much work addressing expectation-guided image analysis [15, 22, 14, 4], but to our knowledge few vision systems with a generic architecture have been proposed which allow to feed back expectations from aggregates at arbitrarily high levels of abstraction to image analysis procedures at the level of raw images. Nagel [1] has been one of the first to demonstrate with concrete experiments in the street traffic domain that high-level hypotheses about intended vehicle behaviour could in fact be used to influence the tracking unit and thus improve tracking under occlusion.

In this contribution we show how a scene interpretation system based on aggregates, introduced as generic high-level conceptual units in [10, 17, 16], can generate feedback in a generic manner. This is demonstrated by experiments with the fully implemented scene interpretation system SCENIC. One of the core mechanisms of SCENIC is the capability of part-whole reasoning, i.e. the capability of establishing an aggregate instantiation based on evidence for any of the aggregate parts. This allows one to generate strong expectations about further evidence for parts of this aggregate and to feed back these expectations to lower levels. In SCENIC, this feedback process has been extended to control image-analysis processes below the level of recognised objects which provide the input to the high-level interpretation system. To our knowledge, this is the first time that a generic high-level scene analysis system can control low-level processing down to the level of raw images.

Our general approach will be described in Section 2. It is motivated by considering scene interpretation as a model construction task (in the logical sense). As such, an interpretation provides instantiations of the conceptual knowledge base consistent with evidence and any context information which may be available. In principle, SCENIC can deliver all consistent interpretations. In its current state, SCENIC does not use probabilistic information as a preference measure for choosing interpretations but explores the space of logically consistent interpretations by specific interpretation strategies.

After the overview we shall present the processing levels from bottom to top and explain their role in feedback processing. In Section 3 one of the low-level modules of SCENIC will be described, specialised for the current application domain of SCENIC, the interpretation of building facades. The module extracts windows and can be controlled by top-down guidance.

In Section 4 we present the intermediate layer between low-level analysis and high-level interpretation. As a two-way link between low-level and high-level processing, it poses several challenges. Its main function is to match evidence with expectations. Given evidence, it generates hypotheses about likely scene objects, given hypotheses it looks for matching evidence or even initiates low-level analysis to procure evidence.

In Section 5, the functionality of the high-level interpretation system is explained in some detail. The high-level system is generic in the sense that the available interpretation steps can generate all consistent interpretations based on evidence of any degree of completeness, incorporating context information at any conceptual level, and initiating feedback to lower-level image analysis if required.

Section 6 presents the results of the feedback example and provides architecture details about the implemented system SCENIC.

Section 7 finally provides a summary and an outlook on future work. Throughout this contribution we shall present examples providing proof-of-concept and demonstrating the results of the implemented system.

2 General Approach

In this section we explain the rationale for modelling scene interpretation as logical model construction and show how this leads to a system architecture where high-level hypotheses connect to image analysis procedures via a middle layer which mediates between hypotheses and evidence.

It was first shown by Reiter and Mackworth [18] and later elaborated in [20], that, under certain assumptions, scene interpretation can be formulated as a (partial) model construction task. In general, to construct a logical model means to construct a mapping from constant symbols and predicates of a symbolic language into the corresponding entities of a domain such that all predicates become true. In scene interpretation tasks, the domain is usually the real world, the constant symbols denote scene elements, objects and higher-level entities determined by a vision system, and the predicates express class membership and relations for such entities. Part of the mapping is determined by low-level scene analysis which connects symbols to real-world scene entities. The remaining part is constructed in terms of hypotheses about the scene and represented by the corresponding symbols as place-holders. Accordingly, from a logical point of view, scene interpretation is the construction of a symbolic description consistent with conceptual knowledge about the world and concrete knowledge about the scene, the latter comprising sensor-based evidence and context information.

As a consequence of the model construction paradigm, the interpretation process naturally leads to the formation of hypotheses originating from conceptual knowledge rather than evidence. In its extreme form it can be called “controlled hallucination”, a term attributed to the British vision pioneer Max Clowes. In a compositional concept hierarchy, high-level hypotheses

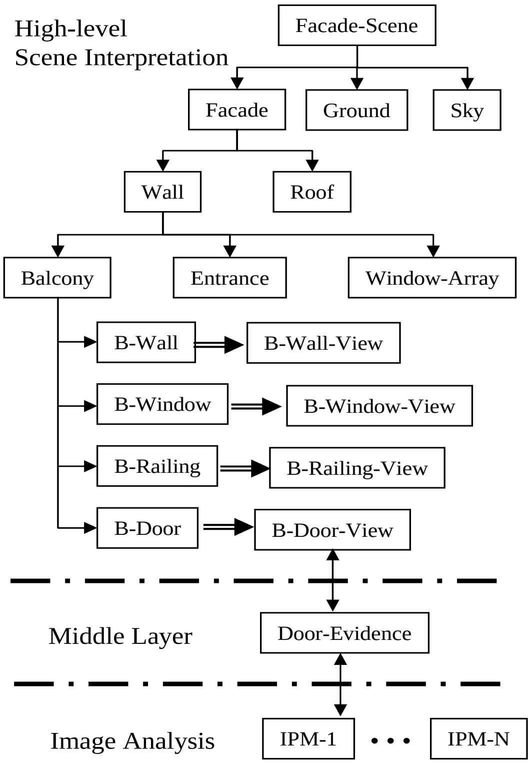

Figure 1: Structure of the scene interpretation system with compositional hierarchy, middle layer and image processing modules (IPMs).

may kick off a cascade of lower-level hypotheses down to the level of symbolic primitives (see Figure 1) - this is the origin of feedback.

The middle layer is what is often called the signal-symbol interface. In our architecture, its task is to generate view hypotheses from evidence generated by image processing modules (IPMs), to assign evidence to view hypotheses generated top-down, and to trigger IMP activities if needed. The capability of the middle layer to control image analysis by high-level hypotheses is at the core of feedback processing in our system architecture.

In the following sections, the three layers of the system architecture will be described in more detail. Examples will be taken from the domain of man-made structures which is studied in the EU-funded project *eTRIMS*. The scene interpretation system SCENIC has the task to interpret images using these learnt concept descriptions.

3 Low-Level Image Analysis

In this section we describe the AdaBoost image processing module (IPM) as an exemplary low-level IPM for generating symbolic scene primitives and accepting top-down requests for feedback processing.

The AdaBoost IPM uses Haar-like features in scale-space to train classifiers to detect appearances of different objects from the domain ontology. It works on rectified grey-scale images and provides confidence-rated bounding rectangles for all detected objects in the image. The classifiers are trained using supervised learning, where positive examples are shown to the system and negative examples are automatically generated. Thus, the typical appearance of an object is learnt and can be used to detect other objects of similar appearance.

The detector can be controlled by several parameters: region of interest (ROI), sensitivity and image scale. The ROI limits the processing to part of the image. This cuts down the processing time, but it can also result in a different alignment of the scale space as compared to unconstrained processing, giving a different set of Haar features. The sensitivity parameter can be changed to allow for accepting weaker evidence. This is especially useful alongside a ROI setting based on a high-level hypothesis. The image scale parameter is a rough measure of how large the objects in the image are, and can be used to filter out results which are too small or too large.

The AdaBoost IPM features a standard interface provided by SCENIC, so it can be replaced by, or used with, other image processing modules offering different functionality.

Experiment We used an AdaBoost-trained detector for T-style windows. The output of the classifier is shown in Figure 2. The results of the low-level detection are passed on to the middle layer as a basis for a symbolic description of the image. Further down, we shall describe how the input from the high-level interpretation module is passed down to the AdaBoost detector to refine its results and find previously undetected objects.

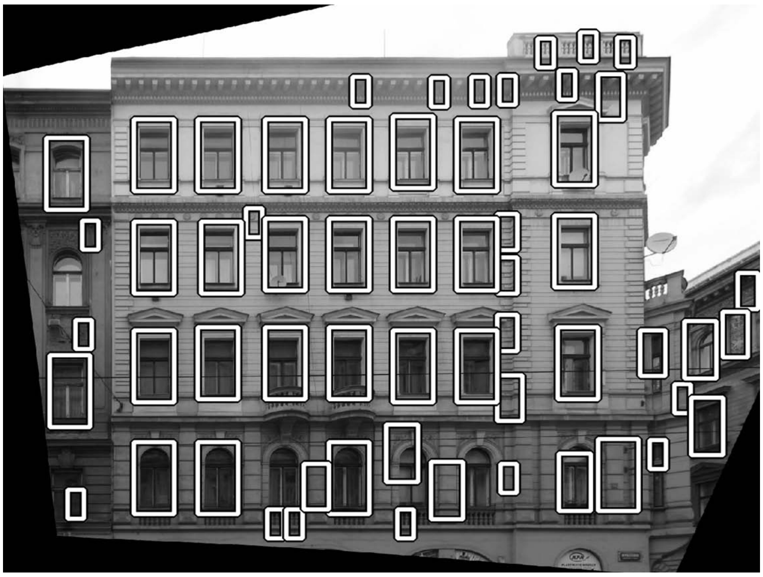

Figure 2: Evidence created with the AdaBoost Image Processing Module.

4 Middle Layer

The middle layer acts as an interface between the image analysis and interpretation layers. Its main tasks are generating symbolic primitives from low-level evidence, confirming or refuting high-level hypotheses, and initiating low-level activities from high-level hypotheses. The combination of these tasks allows the scene interpretation system to operate in a feedback loop as originally proposed by Kanade [13] almost three decades ago.

Providing symbolic input to the interpretation layer is a bottom-up step which matches low-level inputs (referred to as "evidence") to instances of high-level concepts in the knowledge base (referred to as "views"). Currently, the AdaBoost IPM provides pre-classified evidence, with classes from the domain ontology. The middle layer, however, has a more general design and also accepts *typified* evidence (based on shape, colour and patch-based evidence) which is then mapped into views of object classes.

Confirming and refuting high-level hypotheses is a mediating step which takes a hypothesis from the interpretation layer and attempts to match it to available evidence, possibly taking occlusion into account. The matching is done based on the position, size, shape, colour, suggested object class from the low-level, and any further information available about the hypothesis in the knowledge base. The availability of this information depends on the choice of the low-level module. In the case of the AdaBoost detector described in the previous section, the matching is based on the position, size and the object class suggested by the IPM (i.e. window).

If there is sufficient evidence for the existence of the hypothesised object at the given position in the image, the hypothesis is confirmed, and the evidence features are passed on to the interpretation layer. In the case of *negative evidence* (e.g. in terms of a wall where a window is expected) the hypothesis is refuted. If this happens, a conflict is generated in the interpretation layer and backtracking occurs.

If a hypothesis cannot be confirmed or refuted based on the available evidence, the low-level image analysis algorithm can be run again, with the ROI provided by the hypothesis and possibly changed parameters in order to look more closely for partial or weak matches. This is the feedback loop which can find new evidence in the image with the help of high-level predictions.

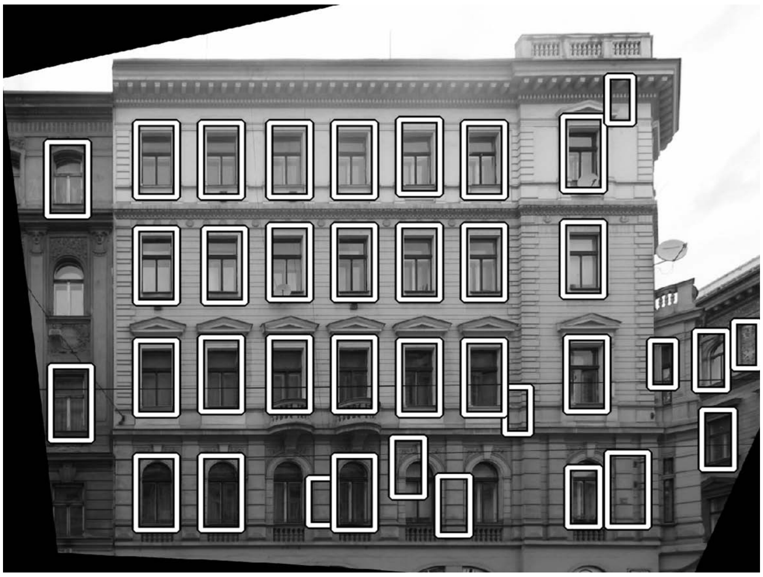

Figure 3: Views created from filtered evidence.

Experiment In our experiment, the middle layer receives evidence from the AdaBoost window detector and creates views from evidence items with a high confidence value. It then passes these views to the interpretation layer (see Figure 3). Once the interpretation layer creates its own view hypotheses, the middle layer tries to match these to existing evidence. If there is insufficient evidence to confirm a hypothesis, it starts low-level image analysis with updated parameters and a specific ROI. The new evidence is used to update features for the hypothesised views, which then are passed back to the interpretation layer, closing the loop.

5 High-Level Interpretation

The goal of high-level interpretation is to create a semantic description of the scene. It uses two inputs: (i) the knowledge base consisting of concepts about possible aggregates and their parts, including constraints among them, and (ii) views created by the middle layer from evidence delivered by image processing modules. The goal of the high-level interpretation process is to instantiate concepts of the knowledge base (i.e. to create a *model*), by checking for which concepts the relevant attributes and constraints are fulfilled.

5.1 Conceptual Knowledge

The conceptual knowledge base implicitly represents all scene interpretations. Each concept is described by attributes with value ranges or value sets, constraints between attributes and relations to other concepts. Spatial attributes representing the potential position and size of the scene objects in terms of ranges are of primary importance. They are represented as constraints which are automatically adjusted as new evidence or related hypotheses restrict the attribute values.

All concepts are organised in a compositional hierarchy where aggregate concepts are related to their parts and vice versa, down to the level of symbolic primitives. Typical aggregates in our domain are `balcony` and `window-array`; typical primitives are `railing` and `door`. Currently, such concepts are learnt from negative and positive examples. Concepts are also organised in specialisation hierarchies.

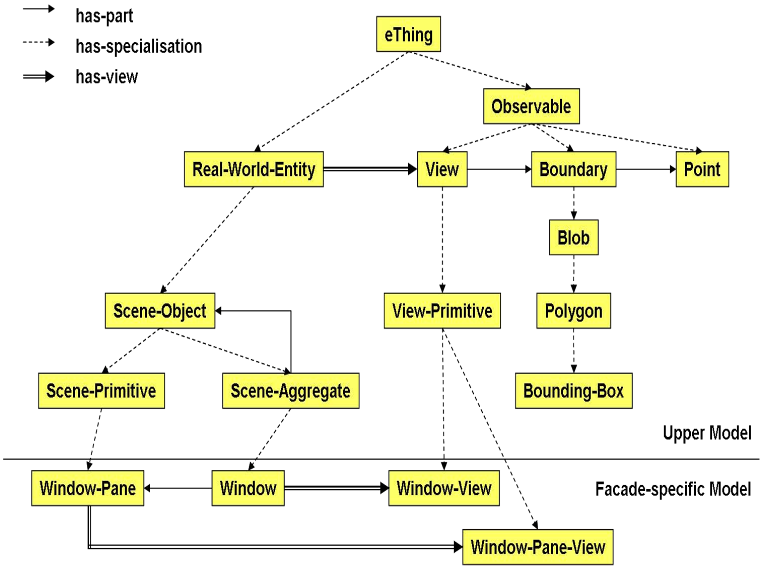

Figure 4: SCENIC upper model with observables (right) and real-world entities (left), both are connected via the `has-view` relation and further structured via `has-part` and `has-specialisation`.

The types of concepts for describing scenes are represented in an *upper model* and domain-related concepts in a *domain-specific model*, which is a model of building facades in our experiment (see figures 4, right). The upper model is structured into two parts: concepts representing real-world entities (`real-world-entity`) and concepts representing the evidence objects generated by sensory equipment (`observables`).

Concepts `real-world-entity` and `view-concepts` are related through the relation `has-view`. For single images, this is a one to one relation which connects the conceptual description of a real-world entity, e.g. a `window`, to the conceptual description of its view, e.g. a `window-view`. This relation also maps between world coordinates and image coordinates.

Experiment In our experiment window arrays are detected using the `window-array` concept which is specified by the following information:

```
Number of parts is [3 to inf].
All parts have type window.
All parts have similar y-position.
Any two neighbouring parts have similar distance.
All parts have similar height.
```

5.2 The High-level Interpretation Process

During the high-level interpretation process concepts have to be instantiated and possible values have to be selected for each attribute or relational parameter. This is done automatically through constraint propagation or by calling compute procedures.

The order of these decisions is crucial for an efficient search process. It is controlled by declaratively represented control knowledge, lacking probabilistic guidance in the current state of implementation. The order of activities induced by the control knowledge is given below for one feedback cycle and illustrated with the window array example.

1. Selecting a top-level scene aggregate A scene aggregate (e.g. `facade-scene`) is selected interactively as a goal of the interpretation process. An interpretation will be completed if instantiations for all objects in obligatory relations to this aggregate are determined, in particular for the descendants of the compositional hierarchy. The interpretation system is also initialised with information relating image coordinates to world coordinates.

2. Starting from given views in a bottom-up manner An appropriate real-world entity is created for each given view following the relation `view-of`. Furthermore, all aggregates are created in a bottom-up manner which follow uniquely from given real-world entities. In this phase, instances of the `window-array` concept are created, when three windows exist which fulfil the `window-array` constraints.

3. Top-down generation of hypotheses The aggregates created in Phase 2 trigger the creation of hypotheses for not yet instantiated parts (other aggregates or primitive objects). For example, if window arrays have been identified which can be enhanced with windows in addition to already instantiated windows, such new windows are searched at appropriate positions inferred from the established windows. When searching for parts, established parts are preferred for integration and new parts are hypothesised only if needed.

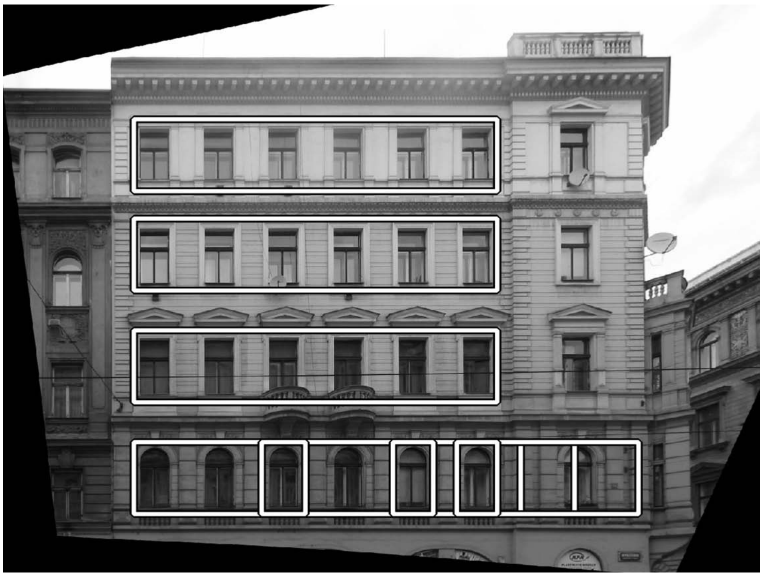

Figure 5: Hypothesised window-arrays with four additional window hypotheses.

4. Backtracking If constraints for a hypothesised entity cannot be fulfilled, a conflict occurs. It is resolved by backtracking to a certain decision made ealier and changing that decision. For example, if it turns out that a hypothesised window cannot be within the computed position range, e.g. because there is no facade, the system can backtrack to the decision where the window had been hypothesised.

5. Hypothesising views After inferencing hypothesised objects in the preceding phases has ended, views with image coordinates are created for each hypothesised scene object and passed to the middle layer for further processing.

Experiment Figure 3 shows the views which are the input for the high-level interpretation. From this input, corresponding instances of type window are created. Next, the interpretation process tries to identify window arrays. When three windows fulfil the constraints of the window array concept, a window-array hypothesis is instantiated. If there are gaps in the window array, which can be filled with further windows, i.e. if $gap < height + 2 \times distance$, new window hypotheses are created. The example shows four new window hypotheses created this way. They reflect the conceptual knowledge that, if several windows with the same size, y-position, and distance are established by an IPM and there are gaps of the proper size, then there should be further windows (see Figure 5).

6 Final Feedback Results and Software System

The high-level interpretation process creates hypotheses about potential objects including their properties, like position and shape, represented by uncertainty ranges and sets of alternative values. These hypotheses describe real-world objects and their real-world coordinates, which need to be mapped onto the image by creating corresponding view hypotheses. These expected views provide feedback for image analysis and restarting the IPMs. In our experiment, hypothesised windows of a window array have the effect of restarting the AdaBoost IPM with changed sensitivity options and focussed on specific ROIs, resulting in new evidence for potential windows. The feedback results are shown in Figure 6: three new correct window evidences in regions where no window has been recognised in the first run of the IPM (first three from left) and one new false window evidence in the wall area on the right. This feedback result confirms all window hypotheses generated by high-level interpretation. It would require an improved window IPM or an additional IPM for recognising walls to correct this low-level mistake.

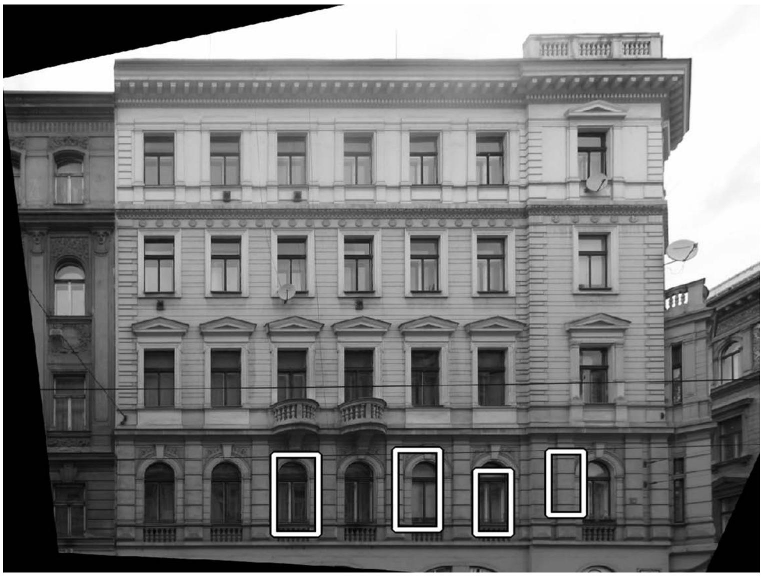

Figure 6: Result after restarting the low-level IPM with a hypothesised region of interest and increased sensitivity.

This example of feedback between high-level interpretation and low-level analysis demonstrates the main strength of this approach. Two kinds of knowledge sources join forces: the low-level IPMs which incorporate knowledge about interesting evidence (e.g. window evidence), and the high-level interpretation system with its knowledge about interesting aggregates (e.g. window arrays). In our implementation, the feedback cooperation of these knowledge sources can be compared to a dialogue, for example:
Low-level: “Here should be a window w of size x.”
High-level: “To the left and right of w are larger windows of size y, thus w should also have size y.”

Software System The implemented system SCENIC consists of five system parts connected via remote procedure calls[1] (see Figure 7). This enables us to plug in different low-level algorithms and allows for distributed operation among several computers in a network.

[1] *www.xmlrpc.com*

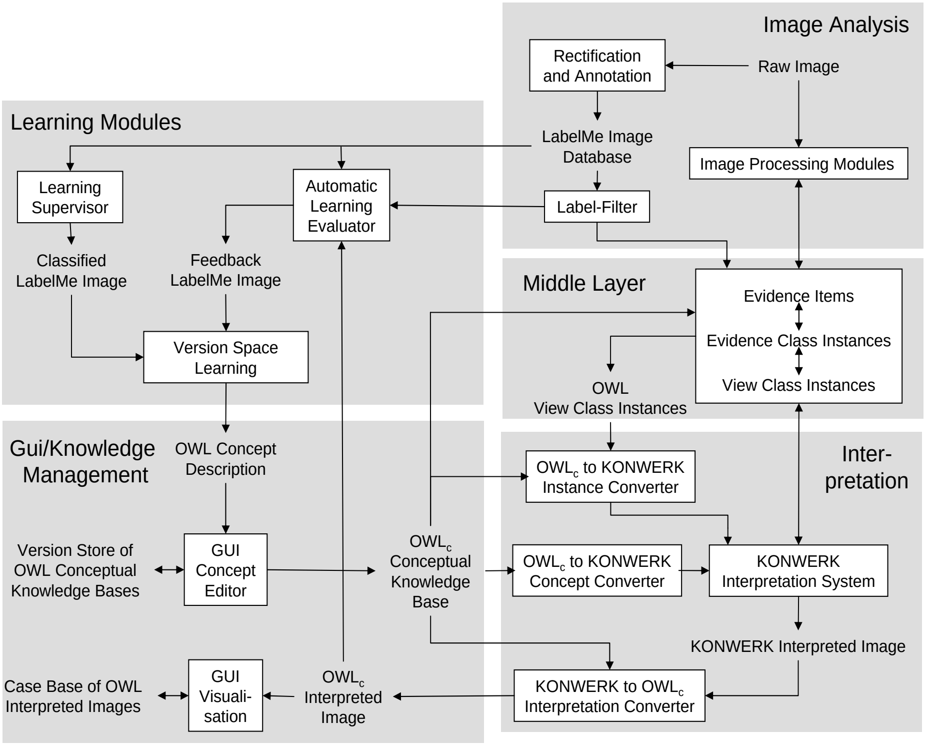

Figure 7: Overview of SCENIC's modules.

GUI/Knowledge Management The GUI has the following tasks: interactive control of the processing levels (image analysis, middle layer, high-level interpretation); presentation and depiction of results; management of distinct versions of the knowledge base. The knowledge base is implemented as an OWL[2] knowledge base and augmented with constraints (depicted as OWL_c in Figure 7). The constraints are represented in a proprietary constraint language [8] enabling n-ary constraints on the concept level.

Image Analysis Image analysis can be performed with distinct types of IPMs or manually by annotating images. Currently we use the IPM described

[2] Web Ontology Language, $www.w3.org/TR/owl-ref/$

in Section 3 and manually annotated LabelMe images[3]. The results of image analysis are represented using an extended LabelMe format.

Middle Layer The middle layer has access to the conceptual knowledge base in order to map IPM output to views which are individuals of view concepts defined in the conceptual knowledge base. The middle layer has also access to the repertoire of IPMs and can select an IPM based on the type of evidence which is required.

Interpretation The interpretation module converts the OWL knowledge base and the input received from the middle layer into internal representations of the structure-based configuration system KONWERK [21, 11, 9, 8, 3], which is reused here for scene interpretation. KONWERK features an expressive concept language, a declarative control language, and inference capabilities based on specialisation relations and a powerful constraint system. An interpretation result can be returned as an image alongside instantiated concepts.

Learning The learning module is a separate module. It provides aggregate concepts in the form of augmented OWL concepts.

7 Summary and Future Work

In this deliverable we have presented a generic approach for combining low-level image analysis and high-level interpretation so that feedback to low-level analysis can be achieved. To this end, a middle layer has been introduced which mediates between low-level evidence and high-level hypotheses. The combination of low-level, middle-layer and high-level techniques has been implemented in the system SCENIC. Results demonstrating the use of feedback have been presented on an image with buildings. The potential of feedback supported by this architecture, however, goes beyond the situation demonstrated by the example. Feedback could help to implement a coarse-to-fine

[3] *labelme.csail.mit.edu*

strategy, with fine-grained analysis only where required, or it could allow image analysis bounded by a high-level focus.

In future work, we shall integrate further IPMs for recognising other primitive objects, e.g. window-panes, but also unclassified evidence which will then have to be classified by invoking high-level knowledge. The conceptual knowledge base will be further extended by learnt concepts. Furthermore, a probabilistic model for the compositional concept hierarchy will be integrated as a preference measure for interpretation steps.

References

[1] M. Arens, A. Ottlik, and H.-H. Nagel. Using Behavioral Knowledge for Situated Prediction of Movements. In *Proc. 27th German Conference on Artificial Intelligence (KI-2004)*, volume LNAI 3238, pages 141–155. Springer, September 2004.

[2] M. Borg, D. Thirde, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, and M. Thonnat. A Real-Time Scene Understanding System for Airport Apron Monitoring. In *Proc. of IEEE International Conference on Computer Vision Systems ICVS06*, 2006.

[3] R. Cunis, A. Günter, and H. Strecker (Hrsg.). *Das PLAKON-Buch*. Springer Verlag Berlin Heidelberg, 1991.

[4] E. Dickmanns. Expectation-based Dynamic Scene Understanding. In A. Blake and A. Yuille, editors, *Active Vision*, 1993.

[5] 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*, 2006.

[6] R. Gerber and H.-H. Nagel. 'occurrence' Extraction from Image Sequences of Road Traffic Scenes. In L. van Gool and B. Schiele, editors, *Proceedings Workshop on Cognitive Vision*, pages 1–8. ETH Zurich, Switzerland, 2002.

[7] S. Gong and H. Buxton. Understanding Visual Behaviour. *Image and Vision Computing*, 20(12):825–826, 2002.

[8] A. Günter. *Wissensbasiertes Konfigurieren.* Infix, St. Augustin, 1995.

[9] A. Günter and L. Hotz. KONWERK - A Domain Independent Configuration Tool. *Configuration Papers from the AAAI Workshop*, pages 10–19, July 19 1999.

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

[11] L. Hotz, K. Wolter, T. Krebs, S. Deelstra, M. Sinnema, J. Nijhuis, and J. MacGregor. *Configuration in Industrial Product Families - The ConIPF Methodology.* IOS Press, Berlin, 2006.

[12] Richard J. Howarth and Hilary Buxton. Conceptual descriptions from monitoring and watching image sequences. *Image and Vision Computing*, 18(2):105–135, 2000.

[13] T. Kanade. Region Segmentation: Signal vs. Semantics. In *Proc. International Joint Conference on Pattern Recognition (IJCPR '78)*, Kyoto, Japan, 1978.

[14] M. Mohnhaupt and B. Neumann. On the Use of Motion Concepts for Top-Down Control in Traffic Scenes. In *Proc. ECCV-90*, pages 598–600. Springer, 1990.

[15] H.-H. Nagel. Natural Language Description of Image Sequences as a Form of Knowledge Representation. In *23. Fachtagung für Künstliche Intelligenz (KI99)*, pages 45–60. Springer, 1999.

[16] B. Neumann and R. Möller. On Scene Interpretation with Description Logics. In *Cognitive Vision Systems*, volume LNCS 3948, pages 247–275. Springer, 2006.

[17] B. Neumann and T. Weiss. Navigating through Logic-based Scene Models for High-level Scene Interpretations. In *3rd International Conference*

on Computer Vision Systems - ICVS 2003, pages 212–222. Springer, 2003.

[18] R. Reiter and A. K. Mackworth. A logical framework for depiction and image interpretation. *Artificial Intelligence*, 41(2):125–155, 1990.

[19] K. Sage, J. Howell, and H. Buxton. Recognition of Action, Activity and Behaviour in the ActIPret Project. *Künstliche Intelligenz*, 3:30–33, 2005.

[20] C. Schröder. *Bildinterpretation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bildern (A theory for computer-based image analysis)*, volume DISKI 196. infix, 1999.

[21] T. Soininen, J. Tiihonen, T. Männistö, and R. Sulonen. Towards a General Ontology of Configuration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), 12*, pages 357–372, 1998.

[22] J.M. Tenenbaum and H.G. Barrow. Experiments in Interpretation Guided Segmentation,. *Artificial Intelligence Journal*, 8(3):241–274, 1977.

[23] M. Vincze, W. Ponweiser, and M. Zillich. Contextual Coordination in a Cognitive Vision System for Symbolic Activity Interpretation. In *Proc. of IEEE International Conference on Computer Vision Systems ICVS06*, 2006.