Division of Work During Behaviour Recognition - The SCENIC Approach

Kasim Terzić[1], Lothar Hotz[2], and Bernd Neumann[1]

[1] Cognitive Systems Laboratory, Department Informatik, Universität Hamburg
22527 Hamburg, Germany
{neumann|terzic}@informatik.uni-hamburg.de
[2] HITeC e.V. c/o Department Informatik, Universität Hamburg
22527 Hamburg, Germany
hotz@informatik.uni-hamburg.de

Abstract. Behaviour recognition in a video scene consists of several distinct sub-tasks: objects or object parts must be recognised, classified and tracked, qualitative spatial and temporal properties must be determined, behaviour of individual objects must be identified, and composite behaviours must be determined to obtain an interpretation of the scene as a whole. In this paper, we describe how these tasks can be distributed over three processing stages (low-level analysis, middle layer mediation and high-level interpretation) to obtain flexible and efficient bottom-up and top-down processing. The approach is implemented in the system SCENIC and currently applied to two domains: dynamic indoor scenes and static building scenes. We include details of an experiment where an ongoing table-laying scene is recognised.

1 Introduction

1.1 Application domains and requirements

Computer Vision in its most general form has been likened to silent-movie understanding [1], where people employ extensive common-sense knowledge about the physical world, typical situations, behaviour of people, and aspirations of individuals. On the first glance, behaviour recognition - which is addressed in the paper - appears to be a more restricted topic, with a focus on the recognition of very specific behaviours such as vandalism in a subway station [2], thefts at a telephone booth [3], filling up at a gas station [4], identifying activities at an airport [5] or placing dishes onto a table [6].

But at the moment one attempts to find a generic framework for behaviour recognition, one faces most of the challenges of silent movie understanding. So what are the challenges of generic behaviour recognition? In the following we propose eleven requirements which go beyond traditional single-object recognition and must be met by a system for behaviour recognition. The requirements pertain to a framework for *model-based* behaviour recognition, i.e. behaviour recognition based on explicit representations of behaviour concepts and the necessary procedures for recognising instances of such models in a concrete scene.

R1 Behaviours describe a scene at an abstraction level above the level of single-object trajectories, requiring qualitative and symbolic representations.

R2 Behaviours are typically embedded in a compositional hierarchy with increasing abstraction towards higher levels.

R3 Behaviours are often defined in terms of qualitative spatial relations between objects. These relations must be evaluated efficiently to support behaviour recognition.

R4 Similarly, behaviours may be defined in terms of temporal relations between parts which also must be evaluated efficiently.

R5 With behaviour recognition, we often face the task of interpreting scenes incrementally and in real-time along the temporal dimension.

R6 Behaviour recognition often involves part-whole reasoning, in particular guessing future behaviour from past observations.

R7 Part-whole reasoning and guessing the future means hypothesising interpretations and hence entails uncertainty management and the need for hypothesis revisions.

R8 Expectations generated by behaviour recognition provide a focus of attention and induce top-down guidance for further processing steps.

R9 Behaviour recognition may require that contextual information from other sources than the visual sensors be exploited.

R10 For behaviour recognition it may be necessary to resort to commonsense knowledge, beyond the knowledge about visual phenomena.

R11 Representation and interpretation facilities of the behaviour recognition framework must be domain independent and adaptable to specific application domains by declarative specifications.

Let us consider an example of the traffic domain to illustrate these requirements. A driver assistance system equipped with a front-view camera is supposed to warn the driver when a person is likely to enter the lane in front of the car. "Entering the lane in front up the car" is a qualitative concept (R1). It may be part of more complex behaviours, such as a pedestrian crossing a street or a child running after a ball (R2). "person on lane in front of car" as well as the behaviour described by "enter" involve qualitative spatial relations (R3). To recognise dangerous situations, the temporal relation between the expected car position and a person on the lane must be determined (R4). This must happen in real-time, keeping up with the evolving scene (R5). If a ball is observed running into the lane, this may be part of a possible event "child running after ball" and should cause a warning (R6). Depending on further circumstances (e.g. a clear view of the curb area), the hypothesis of "child running after ball" may be discarded (R7). A verification of this hypothesis may require focussed image analysis in an area where the child would be expected (R8). Context information, e.g. communicated from another car, may be available and must be considered (R9). The example "child running after ball" also illustrates a simple case of using common sense (R10). A more sophisticated warning system would, for example, also consider a possible fencing which would prohibit a child to enter the lane (R10). Finally, the same framework should be utilisable for - say - behaviour recognition in an elderly-care scenario (R11).

We suggest that a computer vision system for behaviour recognition should be designed to support these requirements as far as possible, and that claims regarding generality should be measured against these requirements. Of course, for specific tasks, it may be appropriate to devise special approaches. But in the interest of economical application developments there is a premium on reusable frameworks meeting all of these requirements.

In this paper we describe our approach towards generic behaviour recognition. In agreement with other existing system frameworks [4, 7–9] and conceptual studies [10, 11], our system consists of three major blocks as shown in Figure 1.

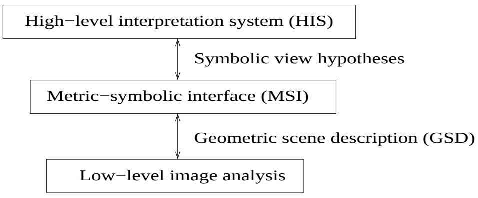

High−level interpretation system (HIS)
Symbolic view hypotheses
Metric−symbolic interface (MSI)
Geometric scene description (GSD)
Low−level image analysis

Fig. 1. Basic system structure for behaviour recognition

Low-level image analysis encompasses diverse image processing modules (IPMs) which compute a geometric scene description (GSD) in terms of segments, blobs or regions of interest (ROIs) tracked through the image sequence. The output is represented in terms of evidence objects which possess both a symbolic identity and a quantitative description. IPMs may be focussed or parametrised by top-down information.

The middle layer is called Metric-Symbolic Interface (MSI), but it has more than a mere interface function. One novel task arising from hypothesis generation in the high-level interpretation unit (R7) is to match top-down hypotheses with bottom-up evidence. This task differs from conventional bottom-up interpretation as (uncertain) hypotheses must be mapped into available evidence or may even trigger IPMs to provide further evidence. Another novel task of the MSI is related to the computation of spatial relations which play a significant role in high-level interpretations (R3). Qualitative spatial relations such as ”touch” or ”on” are natural constituents of symbolic high-level concepts, but they are grounded in the quantitative metrics of the GSD and can be computed much more efficiently using a map-based representation rather than the descriptions of symbolic objects. The same is true for temporal relations (R4) such as ”approach” which also benefit from grounding in a metric representation. A dedicated data structure supporting the representation and computation of spatio-temporal relations has been postulated earlier [12, 13]. In our approach this data structure has a natural place in the MSI.

The high-level interpretation system HIS consists of a conceptual knowledge base and interpretation mechanisms. The conceptual knowledge base describes concepts for object categories, occurrences, behaviours and meaningful object configurations using aggregates as generic structures (R2). The underlying idea is that all high-level structures in a scene can be described in a homogeneous way as composite entities with spatially and temporally related parts. This approach differs from scenarios [9], situation-graph trees [14] or other structures which employ different representations at different abstraction levels, e.g. state-transition networks or Markov Chains for action sequences [3]. We believe that a generic framework (R11) should be able to represent arbitrary temporal relations between the sub-parts of a behaviour (and not only state transition sequences), as can be expressed, for example, by Allen's interval relations [15]. In our implemented system SCENIC (see Section 4) we use quantitative temporal constraints which realise a convex subset of Allen's interval relations [11]. The same formalism has also been proposed by [16].

Another advantage of our homogeneous object-oriented knowledge representation is the possibility to integrate behaviour knowledge with other commonsense knowledge (R10). The viability of this perspective was shown in [17], where description logics were investigated as a knowledge representation framework for scene interpretation. Description logics are known to provide the theoretical basis for knowledge representation with the Semantic Web language OWL[3].

The interpretation mechanism provided by our HIS is designed to deal with incomplete evidence - which is natural in evolving temporal scenes (R6) - as well as additional context information which may be available from other sources than low-level image analysis (R9). This flexibility is achieved by abstaining from an inbuilt interpretation strategy and allowing interpretation steps depending on the information on hand, for example conventional bottom-up steps for interpreting evidence as well as top-down steps for predicting future parts of ongoing behaviour or consequences of context information, for hypothesising occluded objects, for computing spatial and temporal relations in the MSI, or even for triggering focussed image analysis (R8). This general use of top-down steps is novel in existing systems. However, the same idea underlies the temporal prediction mechanism in [14].

In the following sections, we will concentrate on those aspects of our approach which we deem most interesting for the behaviour recognition community. In Section 2, we describe the MSI mediating between symbolic and metric representations. Section 3 presents the knowledge representation and interpretation facilities of the HIS. We use examples from the table-laying domain where the task is to recognise actions such as placing a cover on a table as part of various table-laying behaviours. In Section 4 we present a concrete experiment with our scene interpretation system SCENIC. Section 5, finally, concludes the paper with a summary and an outlook on future research.

2 Middle-level Processing

The metric-symbolic interface (MSI) connects the low-level scene analysis (tracking and primitive object classification) with the reasoning system. It takes input

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

from both the low-level process (in terms of a GSD) and from the reasoning layer (in terms of hypotheses and requests). It has two important tasks: performing a spatiotemporal analysis, which turns the GSD into a set of high-level objects and occurrences such as *moves, touches* and *approaches*, and acting as an interface between the low-level image processing modules (IPMs) and the high-level reasoning system. As a part of this interface work, the MSI creates instances of high-level concepts from evidence, matches hypotheses to existing evidence, and passes information between the low-level and reasoning stages, e.g. initiating a focussed image analysis.

2.1 Low-level input

Although low-level video analysis lies outside of the scope of this paper, we will briefly describe the output of the low-level stage needed for the middle layer. This functionality was implemented as a part of a complete interpretation system for the table-laying scenario (see Section 4).

The low-level stage of video interpretation consists of two main steps: tracking of the objects in the scene and their classification. The tracking stage identifies all moving objects in the scene and assigns each primitive object a unique ID which is kept throughout the interpretation process. The image sequence is sampled at (usually regular) time intervals. The position (oriented bounding box) of each primitive object in motion is recorded for each time instant. The result is a quantitative description of the trajectories of all objects in the scene. Depending on the complexity of the domain, these objects may be blobs, regions of interest (ROIs) or at best regions corresponding to complete physical objects.

The appearance of objects carries important clues about possible classifications and primitive objects are pre-classified using one of many low-level classification algorithms. This is only as reliable as the algorithms used, may be ambiguous, and can be rejected by the high-level stage if it conflicts with other information. Nevertheless, classification is important for the initialisation of the high-level interpretation process.

The result of the low-level analysis is a quantitative description of the scene at each observed time point, consisting of a list of all primitive objects present, each described by: object ID, object class detected by a low-level classifier, position (centre of gravity), orientation, the oriented bounding box, and colour.

2.2 Spatiotemporal analysis

The spatiotemporal analysis within the MSI consists of three steps: calculating *perceptual primitives*, computing *qualitative primitives*, and detecting *occurrences* within the scene.

Perceptual primitives In the first step of the analysis, a set of functions is applied to the object properties from the GSD to obtain quantitative measurements for spatial and temporal relations. The results of these functions are called *perceptual primitives.* The intuition behind this step is to derive location- and timepoint-invariant descriptors. Typical perceptual primitives include:

- the rate of change of the position of an object (its velocity),
- the distance between the centres of two objects,
- the rate of change of this distance,
- the horizontal and vertical distance between axis-parallel bounding boxes of two objects, etc.

These primitives are still quantitative, but they are important for the detection of *primitive occurrences* such as a move (change in position), approach (change in distance) and touch/overlap (intersection of bounding boxes).

Qualitative primitives The second step involves a qualitative evaluation of the perceptual primitives. This is done by applying *predicates* to the perceptual primitives which compute a "qualitative constancy" for each time point, for example containment in a specific value range, being below a certain threshold or being approximately zero. Applying these predicates results in *qualitative primitives*, corresponding to notions like near, far, touching, approaching, moving away, stationary, etc. This process is illustrated with five common qualitative primitives:

- *Moving.* If the difference in the position since the last measurement is approximately zero, the object is *stationary*. Otherwise, it is *moving*.
- *Speed.* The speed of the movement can be qualitatively described by applying a threshold predicate on the rate of change of position of the object. The movement can then be described as *slow*, *fast* or other predicates.
- *Orientation.* By dividing the full circle into several intervals, orientation predicates can be defined relative to the image axes to describe whether the object points *forward*, *backward*, *left* or *right*. These predicates can also be applied to other reference axes, like the direction of movement.
- *Touching.* If the bounding boxes of two objects overlap, the objects are assumed to *touch*.
- *Nearing.* If the distance between two objects is decreasing, the two objects are *nearing* each other.

Spatiotemporal occurrences In the third step, *primitive occurrences* are built by combining qualitative primitives into units extending over time intervals of maximal length, and by creating more complex models. Primitive occurrences, such as move, approach or touch occurrences, form the basis for high-level reasoning.

A primitive occurrence is a concept which encompasses one or more qualitative primitives and a maximal time interval during which the qualitative primitives and possibly a certain set of constraints are always fulfilled. A primitive occurrence is defined by a start and end time, by the objects involved and the qualitative primitives which have to be true during this time period.

- *Move* is an occurrence where an object fulfils the *moving* qualitative primitive throughout a time interval. An interval between two successive move occurrences is a *stay* occurrence.

- *Approach* is an occurrence where the qualitative primitive *nearing* holds between two objects throughout a time interval.
- *Pair move* is a move involving two objects moving at the same speed into the same direction at each instant of an interval. An example is a cup on a saucer moving together.
- *Touch* is defined as an interval during which two objects *touch*, as defined in the previous section.
- *Touching move* is a pair move during which both objects *touch*.
- *Transport* is a touching move consisting of an object which can move by itself (e.g. a person or a hand) and an object which can be moved (e.g. a cup or a saucer).

These processing steps turn a quantitative GSD into a set of qualitative occurrences which can be represented symbolically and correspond to notions used in human perception, thus providing a basis for meaningful high-level concepts. Interesting events in many domains can be described using primitive occurrences of this kind, for example a person purchasing a ticket in a subway station: a person approaches the ticket machine, the person touches the ticket machine, the person moves away from the ticket machine, the person approaches a train.

2.3 Spatial and temporal indexing

The calculation of spatial relationships profits from a map-based representation. Looking for a left neighbour of a primitive object, for example, is simply a matter of traversing a corridor in a map containing all primitive objects, instead of performing an expensive comparison with all objects in the scene. The matching of hypotheses to evidence also profits from this type of representation, as the search can be concentrated on the part of the image confined by the hypothesis.

The map-based representation used in SCENIC is a grid dividing the image into rectangular fields. There is a map representing evidence (computed by low-level image analysis) and one representing views (representing hypotheses of the scene interpretation). Each field contains references to all evidence or view objects whose spatial extent intersects with it. Correspondingly, each evidence item and each view has a list of all fields that it covers in the evidence or view map, respectively. Thus, searching evidence for a hypothesis is turned into a simple lookup operation. The fields covered by a hypothesis in the view map are identified, and the corresponding fields in the evidence map contain references to all applicable evidence items that can be matched to the given hypothesis.

Since the GSD enters the middle layer in terms of data based on image frames, temporal indexing from each time point to objects of the GSD and to qualitative primitives is already available. Primitive occurrences, however, extend over intervals and are not naturally included in a frame-based representation. A top-down request asking for an occurrence in a specific time interval would require checking all primitive occurrences and comparing their begin and end times. Because of this, temporal indexing is extended so that each time point also contains a list of the references to all primitive occurrences taking place at this time point.

2.4 Evidence-view mapping

In addition to the spatiotemporal analysis of the GSD, the central role of the middle layer in SCENIC is matching real-world evidence to instances of object views in the high-level interpretation system. In an interpretation process, this matching may occur in two directions: bottom-up by assigning evidence to a view of a high-level object, or top-down by checking a view hypothesis against the available evidence.

The bottom-up case is a classification step which assigns existing evidence to the view class tied to an object class of the conceptual knowledge base. This step is ambiguous in general, as is well-known from single-object classification, and probabilistic guidance may be required for efficiency. As a result of the classification, a view instance (or in short: view) is created. This bottom-up step is typical for initialising the interpretation process.

In a top-down step, the middle layer receives a view hypothesis created by the interpretation system and has the task of confirming or refuting it. To do so, the middle layer can either match the hypothesised views to known evidence (already identified by the low-level system), or start a new low-level process to look for more evidence at the position indicated by the hypothesis. If a hypothesised view is matched to evidence, the hypothesis is *confirmed* and the evidence is linked to the hypothesis. Otherwise, the hypothesis is *refuted*. The reasoning system can take this new information into account.

In both cases, matches between evidence and views are recorded. If a particular match results in a conflict in the interpretation process, it can be withdrawn. Failed matches are also recorded to avoid repeating them in the future.

Due to the amount of raw data involved in the interpretation of even simple scenes, efficient indexing of information is extremely important when trying to match hypotheses to evidence. The spatial and temporal indexing introduced in this chapter significantly reduces the matching complexity by providing fast access to all evidence in individual space and time segments.

3 Reasoning Level

In our framework we view scene interpretation as a compositional task where the observed spatial and temporal occurrences in a video have to be composed into *aggregates* with increasing level of abstraction until a *scene interpretation* according to a given goal is reached. This composition is based on a declarative representation of the knowledge in a conceptual knowledge base, a *conceptual model* of a domain. In principle, this knowledge generically represents all scenes which may occur in a domain. In the table-laying domain, the conceptual model represents scenes about table laying actions that may occur in such a video.

This conceptual model provides the logical basis for the scene interpretations. In the following, we will shortly describe the knowledge representation language (Subsection 3.1), common aggregates for behaviour recognition (Subsection 3.2), knowledge and reasoning about aggregates for representing compositional (Subsection 3.3), spatial and temporal occurrences (Subsection 3.4) and the process of merging objects (Subsection 3.5).

3.1 Knowledge representation language

The knowledge representation language consists of the following facilities:

Concept Hierarchies. Object classes (*concepts*) are described using a highly expressive object description language, and embedded in taxonomic and compositional hierarchies. Object properties are specified by parameters with restricted value ranges or sets of values. A compositional hierarchy is induced by the special structural relation `part-of`. All concepts are compositional structures called *aggregates* except concepts without parts, which are called *primitive aggregates*. Objects selected for a concrete scene interpretation are instantiations of these concepts.

Constraints. Constraints pertaining to properties (relations or parameters) of more than one object are administered by a constraint net. Conceptual constraints are formulated as part of the conceptual knowledge base and instantiated as corresponding objects are instantiated. Constraints are multidirectional, i.e. propagated regardless of the order in which constraint variables are instantiated. At any given time, the remaining possible values of a constraint variable are given as ranges or value sets.

Task Description. A task is specified in terms of an aggregate which must be constructed (the *goal*) and possibly additional restrictions such as choices of parts, prescribed properties, etc. Typically, the goal is the root node of the compositional hierarchy governing the concepts which are relevant for the task.

Control Knowledge. Strategies for controlling the inference process can be specified in a declarative manner. For example, it is possible to prescribe phases of bottom-up or top-down processing conditioned on certain features of the evolving scene interpretation. As mentioned earlier, there is no inherent interpretation strategy built into the system.

This knowledge representation language is logic-based, general and thus, domain-independent. It is used to model knowledge 1) specific for behaviour recognition in general by specifying an appropriate *upper model* and 2) specific for a certain domain, like table-laying scenarios.

3.2 Upper model

The upper model enables the distinction between occurrences and parts of occurrences representing real world entities (i.e. 3D-objects and their behaviour). Views of primitive occurrences are identified by the middle layer and passed to the high-level system. Conceptually, views are instances of the concept `view` (or its specialisations) of the upper model (see Figure 2). The upper model also contains view classes for all distinct primitive occurrences that can be identified by the middle layer (e.g. of the type move, stay, touch, or approach).

3D-objects are instances of the concept `real-world-entity` or its specialisations. A 3D-object instance may be related to a corresponding view object linked to evidence in the scene, or may be hypothesised without evidence [18]. Further upper-model concepts that are specific to behaviour recognition, such as `transport`, `action`, `sub-action`, are discussed in the following subsections.

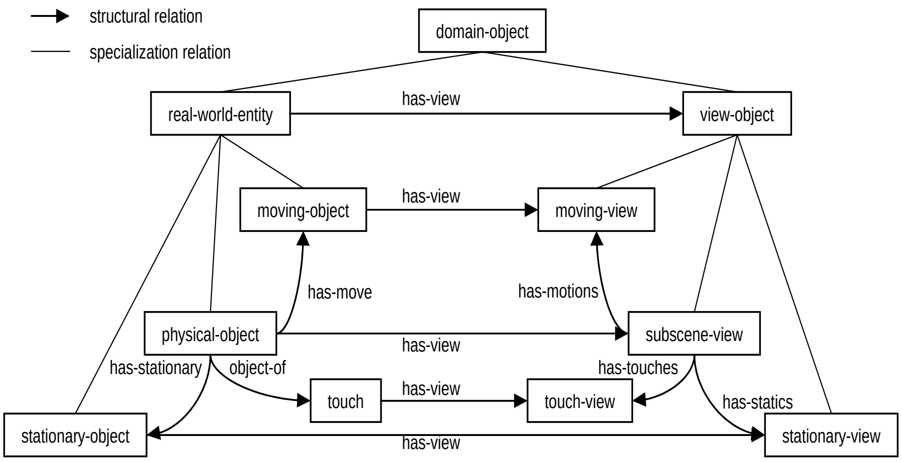


Fig. 2. Upper-Model for interpreting behaviours.

3.3 Reasoning with aggregates - integrating perceived parts and making hypotheses

A scene interpretation may be described as an aggregate composed of behaviours of constituent objects which in turn may be aggregates with constituent parts, etc. In the table-laying scenario, laying a complete cover (`create-cover-action`) may consist of laying a cup cover (`create-cupcover-action`) and laying of a plate cover (`create-basecover-action`). Laying a cup cover consists of transport occurrences of the involved objects (e.g. `hand-cup-transport` and `hand-saucer-transport`), a stationary occurrence representing the steady state of a laid cup cover (`cupcover`) and optionally an approach occurrence prescribing the decrease of distances between the object cup and the object saucer which are part of creating a cup cover (`cup-saucer-approach`).

Parts of an aggregate may be mandatory, optional or number-restricted. An example represented in our knowledge-representation language is given below:

```
(define-concept :name create-cupcover-action
  :super create-action
  :relations
   ((has-cup-transport       (:set (:some (a hand-cup-transport)    :min 1 :max 1)))
    (has-saucer-transport    (:set (:some (a hand-saucer-transport) :min 1 :max 1)))
    (has-spoon-transport     (:set (:some (a hand-spoon-transport)  :min 1 :max 1)))
    (has-cup-saucer-approach (:set (:some (a cup-saucer-approach)   :min 0 :max 1)))
    (has-cupcover            (:set (:some (a cupcover) :min 1 :max 1 )))
    (subaction-of (a create-cover-action))))

(define-concept :name create-cover-action
  :super create-action
  :relations
   ((has-subactions (:set (:some (a create-action) :min 2 :max 2)
                     :specializations
                          (:some (a create-basecover-action) 1 1)
                          (:some (a create-cupcover-action) 1 1)))
    (has-cover (:set (:some (a cover) :min 1 :max 1 )))
    (action-of (:or (a dinner-for-two-si) (a single-dinner-si)))))
```

```
(define-concept :name dinner-for-two-si
  :super cover-interpretation
  :relations
   ((has-actions (:set (:some (a create-cover-action) :min 2 :max 2)))))
```

Such concept descriptions are used to reason about the compositional structure of a scene in a top-down or bottom-up manner. For example, if a `create-cover-action` was instantiated for some reason, the appropriate parts are instantiated top-down (i.e. hypothesised objects are created). If a `hand-cup-transport` was instantiated, it is recognised as part of a `create-cupcover-action` and the corresponding aggregate is instantiated bottom-up. If variability occurs (for example `create-cover-action` can be part of `dinner-for-two-si` as well as `single-dinner-si`), a mechanism is needed for selecting one interpretation and evaluating it. In our approach, we currently use backtracking search, but probabilistic methods are also being developed.

3.4 Spatial and temporal reasoning

Conceptual descriptions of a scene involve spatial and temporal properties of occurrences and spatial and temporal relations between occurrences to a significant extent. The SCENIC approach supports this by providing appropriate concept parameters (like `tp-end`, `tp-start` for time intervals and `bb-left-upper-x` etc. for bounding boxes) and constraints related to these parameters.

When an occurrence is inferred for a video scene, the corresponding concept is instantiated with the spatial and temporal parameters as described above. They are initially set to specific values provided by the middle layer (if created bottom-up) or to intervals provided by the concept definition (which may be the open range of $[0 \ldots inf]$) in the case of top-down hypotheses. By processing the related spatial and temporal constraints in subsequent processing steps, the value ranges of parameters are further reduced, leading to final uncertainty intervals or conflicts causing backtracking.

We distinguish between *conceptual constraints* and *constraint relations.* Constraint relations are equations or inequalities about spatial and temporal parameters. Conceptual constraints describe a structural situation which is a precondition for evaluating certain constraint relations. For example:

```
(define-conceptual-constraint
  :name cup-before-saucer
  :structural-situation
   ((?cupcover-act :name create-cupcover-action)
    (?cup-tp       :name hand-cup-transport
                   :relations ((cup-transport-of ?cupcover-act)))
    (?saucer-tp    :name hand-saucer-transport
                   :relations ((saucer-transport-of ?cupcover-act))))
  :facts ((>= (?cup-tp tp-start) (?saucer-tp tp-end))))
```

specifies that, if a `create-cupcover-action` has a `hand-cup-transport` and a `hand-saucer-transport` (described with the structural situation), then the time point `tp-start` of the `hand-cup-transport` should be after (>=) the `tp-end` of the `hand-saucer-transport`. Conceptual constraints are specified for concepts and hold for every instance of these concepts (here for every `create-cupcover-action`).

Constraints can also be defined domain-independently for concepts of the upper model, e.g. for computing a bounding-box of a real-world entity.

3.5 Merging objects

A further reasoning service is needed when two objects were created independently but can be treated as the same object. In this case both object instances should be *merged* as this is will provide a simpler and hence preferable scene description. The need for a merge may occur, for example, when an object has been hypothesised top-down (say, a laying-a-dinner-for-two-action) and bottom-up processing of evidence has come up with the same hypothesis. Merging implies that the two objects are unified with all their properties and relations. In SCENIC, merging is accomplished by a conceptual constraint specifying all conditions which must be fulfilled by the two merging candidates.

4 System and Experiments

4.1 Architecture

The system SCENIC consists of five system components connected via remote procedure calls and file transfer[4]. This enables us to plug in different low-level algorithms and allows for distributed processing on several computers in a network. In the following we give details about each of the system components.

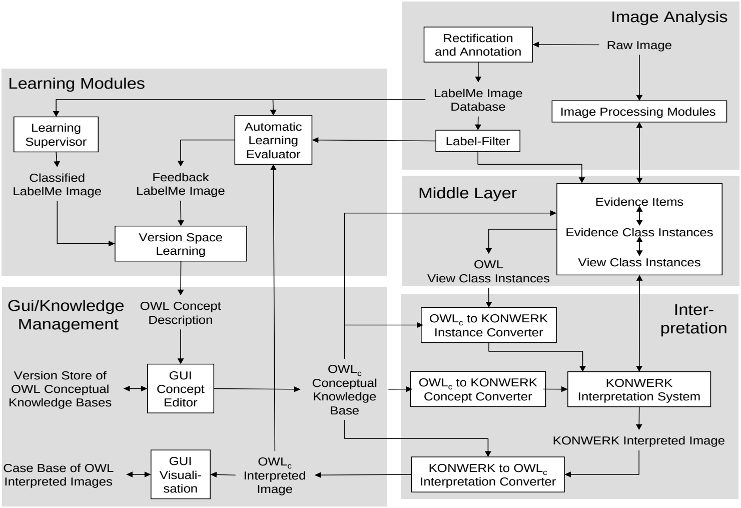


Fig. 3. Overview of SCENIC's modules.

[4] We are currently migrating from a stream-based list format (see below) to an XML-RPC interface (*www.xmlrpc.com*).

GUI/Knowledge Management The GUI has the following tasks: Interactive control of the three processing levels low-level 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 knowledge base and augmented with constraints (depicted with OWL_c in Figure 3). The constraints are represented with a proprietary constraint language [19] enabling n-ary constraints at the concept level.

Image and Video Analysis Image and video analysis can be performed with distinct types of image processing moduls (IPMs) or manually by annotating images. For behaviour recognition we use a simple tracking unit and a model-based object recognition algorithm (see Section 2). The results of video analysis are represented using a proprietary format for the Geometric Scene Description (GSD), see Section 4.2.

Middle Layer The middle layer has access to the conceptual knowledge base in order to map IPM output to views which are instances of view concepts defined in the conceptual knowledge base. For behaviour recognition the middle layer mainly identifies trajectories in the GSD and recognizes move and touch occurrences.

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 [19–23], 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.

Learning The learning module is a separate module not relevant for the topics of this paper (see [24]). It provides aggregate concepts in the form of augmented OWL concepts.

4.2 Experiments

We have executed several experiments with SCENIC in the dynamic table-laying domain [6, 18] and the static building domain [25]. In this paper we focus on the interplay between high-level interpretation and middle layer in a dynamic scene. As input, we use a video where two human agents, sometimes acting in parallel, place dishes and other objects onto a table, for example, create covers as customary for a dinner-for-two. The tracking system identifies primitive objects in each frame, e.g.:

```
(FR 188 (ID 1  (PV TYPE SAUCER)(PV CENTER (435 191))(PV BOX (404 160 467 224))
               (PV SM(20 10 17 0 3 98)))
        (ID 2  (PV TYPE PLATE)(PV CENTER (110 274))(PV BOX (64 228 158 322))
               (PV SM(2 0 3 98 0 5)))
        (ID 3  (PV TYPE UNKNOWN)(PV CENTER (427 379))(PV BOX (411 369 445 387))
               (PV SM(0 0 0 0 0 0))))
        ...
(FR 216 (ID 1  (PV TYPE SAUCER)(PV CENTER (435 191))(PV BOX (404 160 467 224))
        (PV SM(20 10 17 0 3 98)))
(ID 2   (PV TYPE PLATE)(PV CENTER (110 274))(PV BOX (64 228 158 322))
        (PV SM(2 0 3 98 0 5)))
(ID 4   (PV TYPE SAUCER)(PV CENTER (209 311))(PV BOX (178 281 241 344))
        (PV SM(13 5 11 0 1 97)))
(ID 3   (PV TYPE SAUCER)(PV CENTER (437 199))(PV BOX (404 159 467 250))
        (PV SM(12 14 20 9 7 75)))
(ID 54  (PV TYPE HAND)(PV CENTER (211 362))(PV BOX (199 337 224 385))
        (PV SM(0 0 0 100 0 0)))
(ID 53  (PV TYPE HAND)(PV CENTER (477 272))(PV BOX (454 242 500 303))
        (PV SM(0 0 0 100 0 0))))
```

The MSI identifies movements, stationary occurrences and touches for all objects and stores the results in so called *motion frames* (see below). However, not all possible spatial data are initially created. For example, distance changes (i.e. approach occurrences) between all objects are not computed for combinatorial reasons.[5]

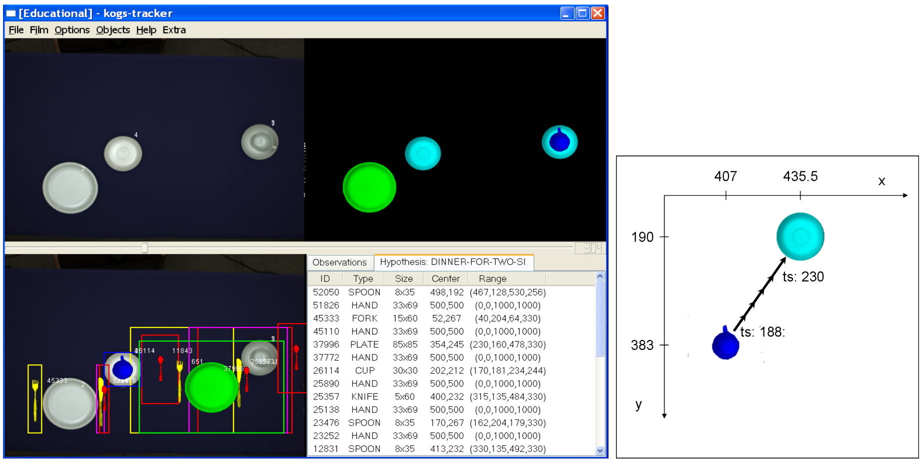
```
MOTION-FRAME:
object: #3 object-types: (SAUCER-VIEW UNKNOWN-VIEW CUP-VIEW)
type: GENERAL-MOTION  start: (-1000000000000 188) end: (228 230)
trajectory: ((-1000000000000 (427 379)) (190 (429 371)) (192 (430 358))
             (194 (431 345)) (196 (432 332)) (198 (433 319))
             (200 (434 307)) (202 (435 295)) (204 (436 284))
             (206 (437 274)) ...)
```

The high-level unit receives the move, stay and touch occurrences in form of instances of `moving-view`, `stationary-view` and `touch-view` as input. The interpretation process uses the conceptual model (see Section 3) as basis for interpreting the scene. In Figure 4 left, an intermediate scene interpretation is illustrated. Besides others, the system has recognized a `create-cupcover-action`. As defined in the model for `create-cupcover-action`, a `cup-saucer-approach` has to be present. The high-level system therefore creates a hypothesis for such an approach object with the appropriate time and spatial parameters, inferred from the transport objects (see Figure 4 right). This approach object is passed to the middle layer as feedback from high-level interpretation. The middle layer computes all approach objects in the given temporal and spatial region of interest and matches the given hypothesis against the computed evidence. It confirms the hypothesis and thus, supports the hypothesized interpretation.

5 Conclusions

In this paper, we have presented the SCENIC approach to video interpretation. This approach features a flexible mix of bottom-up and top-down processing steps and a division of tasks distributed over (i) a low-level stage for image analysis and tracking, (ii) a middle layer for matching evidence with primitive occurrences, and (iii) a high-level interpretation system for composing the scene description. The middle layer has several novel features: It supports selective computation of spatiotemporal relations using top-down guidance and

[5] One might argue that in a case as simple as our experimental table-laying scene, the combinatorial explosion of binary spatial object relations may be ignored. However, we aim at a system architecture which can be scaled up to more complex scenes.


[Educational] - kogs-tracker
File Film Options Objects Help Extra
Observations
Hypothesis: DINNER-FOR-TWO-SI
407
435.5
x
190
383
y
ts: 230
ts: 188:

ID	Type	Size	Center	Range
52050	SPOON	8x35	498,192	(467,128,530,256)
51826	HAND	33x69	500,500	(0,0,1000,1000)
45333	FORK	15x60	52,267	(40,204,64,330)
45110	HAND	33x69	500,500	(0,0,1000,1000)
37996	PLATE	85x85	354,245	(230,160,478,330)
37772	HAND	33x69	500,500	(0,0,1000,1000)
26114	CUP	30x30	202,212	(170,181,234,244)
25890	HAND	33x69	500,500	(0,0,1000,1000)
25357	KNIFE	5x60	400,232	(315,135,484,330)
25138	HAND	33x69	500,500	(0,0,1000,1000)
23476	SPOON	8x35	170,267	(162,204,179,330)
23252	HAND	33x69	500,500	(0,0,1000,1000)
12831	SPOON	8x35	413,232	(330,135,492,330)

Fig. 4. Left: Intermediate scene interpretation as an instance of lay-dinner-for-two. Objects in natural colours are supported by evidence, objects in artificial colours are hypotheses based on high-level conceptual knowledge. Hypotheses are shown at the center of boxes, which represent possible locations. The low-level result is presented at top right; the original video at top left. Right: Hypothesized approach object for cup and saucer.

exploiting its map-based representations, and it evaluates top-down hypotheses by matching a hypothesis against available evidence or even initiating low-level image analysis processes. An experiment has been presented which illustrates feedback in form of a hypothesis from the high-level to the middle layer. Future work will include learnt concepts about scenes and a probabilistic guidance for the selection of interpretation steps.

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. Neumann, B., Novak, H.J.: Event Models for Recognition and Natural Language Description of Events in Real-World Image Sequences. In: Proc. of Fifth Int. Joint Conf. on AI IJCAI-83. (1983) 724–726
2. Vu, V.T., Brémond, F., Thonnat, M.: Automatic video interpretation: A novel algorithm for temporal scenario recognition. IJCAI (2003) 1295–1302
3. Hongeng, S., Bremond, F., Nevatia, R.: Representation and Optimal Recognition of Human Activities. In: IEEE Proceedings of Computer Vision and Pattern Recognition, South Carolina, USA (2000)
4. Nagel, H.H.: From image sequences towards conceptual descriptions. Image Vision Comput. **6**(2) (1988) 59–74

5. Thirde, D., Borg, M., Ferryman, J.M., Fusier, F., Valentin, V., Brémond, F., Thonnat, M.: A real-time scene understanding system for airport apron monitoring. In: ICVS '06: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, Washington, DC, USA, IEEE Computer Society (2006) 26
6. Hotz, L., Neumann, B.: Scene Interpretation as a Configuration Task. Künstliche Intelligenz **3** (2005) 59–65
7. Bauckhage, C., Hanheide, M., Wrede, S., Sagerer, G.: A cognitive vision system for action recognition in office environments. In: CVPR (2). (2004) 827–833
8. Heintz, F., Doherty, P.: DyKnow: A framework for processing dynamic knowledge and object structures in autonomous systems. In: Proceedings of the International Workshop on Monitoring, Security, and Rescue Techniques in Multiagent Systems (MSRAS). (2004)
9. Brémond, F., Thonnat, M., Zuniga, M.: Video understanding framework for automatic behavior recognition. Behavior Research Methods **3**(38) (2006) 416–426
10. Aisbett, J., Gibbon, G.: A General Formulation of Conceptual Spaces as a Meso Level Representation. Artificial Intelligence **133**(1-2) (2001) 189–232
11. Neumann, B.: Natural language description of time-varying scenes. In Erlbaum, L., ed.: Semantic Structures, D. Waltz (1989) 167–206
12. Mohnhaupt, M., Neumann, B.: On the Use of Motion Concepts for Top-Down Control in Traffic Scenes. In: Proc. ECCV-90, Springer (1990) 598–600
13. Gärdenfors, B.: Conceptual Spaces: The Geometry of Thought. MIT Press, Cambridge, MA, USA (2000)
14. Arens, M., Ottlik, A., Nagel, H.H.: Using Behavioral Knowledge for Situated Prediction of Movements. In: Proc. 27th German Conference on Artificial Intelligence (KI-2004). Volume LNAI 3238., Springer (September 2004) 141–155
15. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11) (1983) 832–843
16. Ghallab, M.: On chronicles: Representation, on-line recognition and learning. In: KR. (1996) 597–606
17. Neumann, B., Möller, R.: On Scene Interpretation with Description Logics. In: Cognitive Vision Systems. Volume LNCS 3948., Springer (2006) 247–275
18. Hotz, L.: Configuration Configuration from Observed Parts. In: Proc. of 17th European Conference on Artificial Intelligence (Configuration Workshop), Riva del Garda, Italy (2006)
19. Günter, A.: Wissensbasiertes Konfigurieren. Infix, St. Augustin (1995)
20. Soininen, T., Tiihonen, J., Männistö, T., Sulonen, R.: Towards a General Ontology of Configuration. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), 12 (1998) 357–372
21. Hotz, L., Wolter, K., Krebs, T., Deelstra, S., Sinnema, M., Nijhuis, J., MacGregor, J.: Configuration in Industrial Product Families - The ConIPF Methodology. IOS Press, Berlin (2006)
22. Cunis, R., Günter, A., Strecker (Hrsg.), H.: Das PLAKON-Buch. Springer Verlag Berlin Heidelberg (1991)
23. Günter, A., Hotz, L.: KONWERK - A Domain Independent Configuration Tool. Configuration Papers from the AAAI Workshop (July 19 1999) 10–19
24. Hartz, J., Neumann, B.: Version Space Learning of Spatial Structures for High-Level Scene Interpretation. eTRIMS EU-Project, Deliverable D2.4 (2007)
25. Hotz, L., Neumann, B., Terzić, K., Šochman, J.: Feedback between Low-Level and High-Level Image Processing. In: submitted. (2007)