Control of Scene Interpretation

J. Matas, P. Remagnino, J. Kittler, and J. Illingworth
Dept. of Electronic and Electrical Engineering,
University of Surrey,
Guildford, Surrey GU2 5XH, United Kingdom

1 Introduction

One of the main goals of visual sensing is to interpret the perceived visual data. By interpretation we understand the process of recovering information relevant to the goals of the autonomous system for which the visual sensor acts as one of its intelligent agents. This definition allows us to approach interpretation as a dynamic control problem of optimal resource allocation with respect to a given objective function.

At the level of the symbolic scene interpretation module of the of the Vision as Process (VAP) system, the surrounding environment is modelled as an organized collection of objects. A system goal therefore typically requests information about objects present in the viewed scene, their position and orientation, dynamics, attributes etc.

The main thesis behind the approach to symbolic scene interpretation in the Vision as Process (VAP) system is that spatio-temporal context plays a crucial role in the symbolic scene model prediction and maintenance. Another essential and distinctive feature of the novel approach is the active control of the visual sensor (mobile stereo camera head) based on the given visual goal and the current symbolic description of the scene. At any stage of processing, the spatio-temporal context is used to select the most suitable representation of objects permitting as efficient matching of image-derived data as possible.

The architecture of the scene interpretation module is based on the hypothesis that control actions implied by any visual task fall into three independent categories: active sensor (camera) control, control of the focus of attention (region of interest definition) and selection of the appropriate recognition strategy. The complex dynamic control problem can therefore be decomposed into a sequence of primitive visual behaviours. From the implementational point of view these primitive behaviours can be effected by issuing parameterised canonical control commands to the basic controllable entities of the module. These comprise i)camera next look direction, ii) camera position or zoom (only camera position control is currently available in the VAP skeleton system), iii) region of interest, and iv) knowledge source selection. The commands are implicitly encoded by the system supervisor in terms of the system goal and perceptual intentions.

A second distinctive feature of our approach to scene interpretation is the use of temporal context. Past experience in the form of information about recognised object is organized in a hierarchical database. In continuous interpretation this information is exploited to implement the focus of attention mechanism. Several 'forgetting' schemes are adopted to reflect dynamism of objects.

With all the building blocks of the VAP system in place [1], it has been possible to demonstrate the merit of spatio-temporal context in scene understanding to validate the cornerstone of the VAP philosophy. This paper gives an account of the initial testing of the hypothesis in the setting of a simple table top scene of limited object dynamics. Typical experiments performed involve the verification of the presence of or the pose of a known object using resources commensurate to the information content of the spatio-temporal context established to date.

The presented work draws upon results in a number of research areas: active sensor control [27] [25], selective perception [28], knowledge representation [26], learning [22], integration of knowledge sources [18]. The main novel feature of our approach is the close interaction of the system goal, sensor control and the visual task. More sophisticated individual components of high-level vision systems have been described in literature, eg. the uncertainty calculus used in the VISIONS system [21], the exploitation of geometrical constraints in ACRONYM [20] or the integration of information from multiple views in the system proposed for terrestrial robots by Lawton et al. in [24] [23].

The report is structured as follows. In Section 2 a brief overview of the VAP system is presented. The discussion is centered around the symbolic scene interpretation module; attention is given to the interface to supervisor and image description modules. The primitive visual behaviours which define the system capability are listed in Section 3.5. Section 3 describes the architecture of the Symbolic Scene Interpretation module, together with the main object recognition knowledge sources. Section 5 introduces the scenario adopted and describes the experiments conducted. Conclusions are drawn in Section 5.3.

2 VAP SYSTEM OVERVIEW

The concepts which are central to VAP have been outlined in the previous section. Their realisation was tested within the system architecture illustrated in Figure 1. From the point of scene interpretation the VAP system can be divided into four functional blocks. These blocks encapsulate the processes that transform lower level descriptions, into more abstract descriptions and correspond to fairly conventional ideas concerning levels of representation in a vision system (i.e.images $\rightarrow$ 2D primitives (lines, ellipses, curves, perceptual groupings, etc) $\rightarrow$ objects). Top-down flow of control information, also depicted in Figure 1, implements the mechanism of focus of attention. In addition, individual modules maintain temporally evolving models (either implicitly or explicitly) of the aspects of the world that they understand. These models are part of the mechanism for exploitation of context.

The lowest level input to the system is provided by the sensor. This is a limited resource with several controllable parameters including position, look direction, aperture, focus etc. Its parameters can be controlled by any other module; the system supervisor acts as an arbitrator of conflicting requests.

The image description module transforms image data into 2D percepts such as edges, lines, elliptical arcs, perceptual groupings etc. . The processing is carried out over several scales on all data currently available to the sensor and the internal model maintained by the module is constituted by the intrinsic parameters of the processes at this level. These parameters can be adjusted by top-down signals from higher levels if the module is consistently found to produce output which is not useful to the higher level modules.

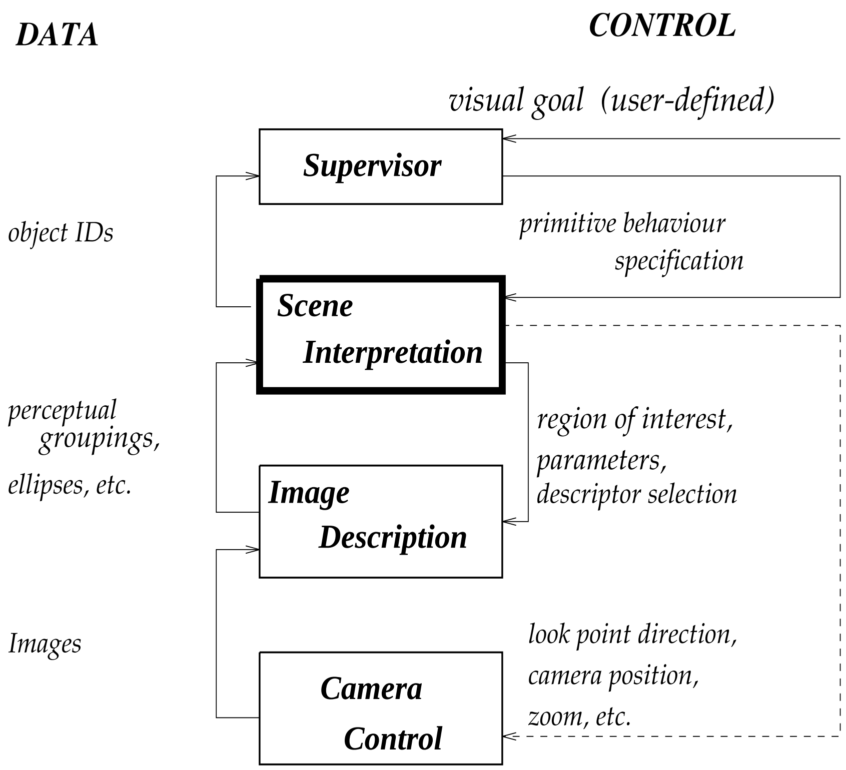

Figure 1: Simplified VAP system architecture. Details of the Scene Interpretation module are depicted in figure 2.

The function of the scene interpretation module is discussed thoroughly in section 3; basically, the module accesses the 2D description and produces an attributed symbolic model which describes the type and pose of identified objects. The model is maintained over a larger spatial and temporal field than that considered by the sensor and the existing partial world model is the context used to select among the possible solutions available for goal satisfaction.

The top level module is the interface and supervisor which connects the system to the external world and which arbitrates requests from other modules for selective attention to be given to parts of that external world. It has reasoning capabilities to transform tasks into the visual goals that other modules can understand.

Overall, the system organisation is hierarchical and functions in a perceptual cycle where goals and parameters are given to a module and the results of this processing are compared to expectations or previous results with the differences being used to update models and guide subsequent actions.

3 Symbolic Scene Interpretation Module Architecture

An efficient use of resources for supervisor goal satisfaction poses a difficult dynamic control problem. In the process of the interpretation module design we observed that organizing resources into three basic operational units, *camera strategy unit, region of interest unit* and *the recognition unit* communicating via *the scene model database* significantly simplified the mapping of supervisor goals into control strategies.

The functionalities of the individual units are independent and complementary; it is therefore possible to factorise the system goal into a combination of a limited set of canonical parameterised control commands to the camera, region-of-interest and recognition units. Besides that, the appeal of the proposed structure stems for the fact that each of the modules corresponds to a well established high level concept. Based on the system goal and current knowledge about the surrounding environment, the camera strategy unit attempts to position and direct the sensor to simplify recognition (in accord with the paradigm of *active vision*); the region of interest unit selects for processing only relevant parts of acquired data (implementing the *focus of attention* mechanism). Information about recognised objects (the system 'history' or 'experience') is maintained in the scene model database enabling the other units to exploit *temporal context* in their operation.

Within this *distributed* framework, the central controller of the interpretation module is virtually nonexistent, its functionality degenerating into:

- passing the appropriate part of the supervisor goal in the form of a control command to individual units
- synchronisation of operation of the continuously operating units

The interpretation module structure is depicted in figure 2. The interpretation process is continuous; the loop in the centre of figure 2 presents all actions repeatedly taken to process consequtive images. The camera strategy, region-of-interest and recognition unit change their mode of operation according to the respective part of a supervisor goal. Possible modes of operation are listed in boxes next to the main loop. An example of an operational mode for each of the basic units can be found inside the block representing the unit.

The in-depth description will proceed as follows. First, the basic operational units - the scene model database, camera strategy unit, region of interest unit and the recognition unit, will be described. Second, modification of the module operation as a response to various supervisor goals will be discussed. Finally, the continuous interpretation cycle, *the loop of perception*, will be presented.

3.1 The Scene Model Database

Exploitation of temporal context in scene interpretation through accumulation of information about objects recognised during the lifetime of a continuously operating vision system allows for a gradual improvement of performance. As the amount of information about the surrounding environment increases, more and more supervisor goals can be satisfied by database queries and/or spatially focused (and therefore efficient) visual processing.

In the scene model, objects are characterised by their recognition class (type), pose (ie. position and orientation), spatial extent and mobility. A confidence value is attached to each attribute. The choice of object attributes reflects the needs that the scene model

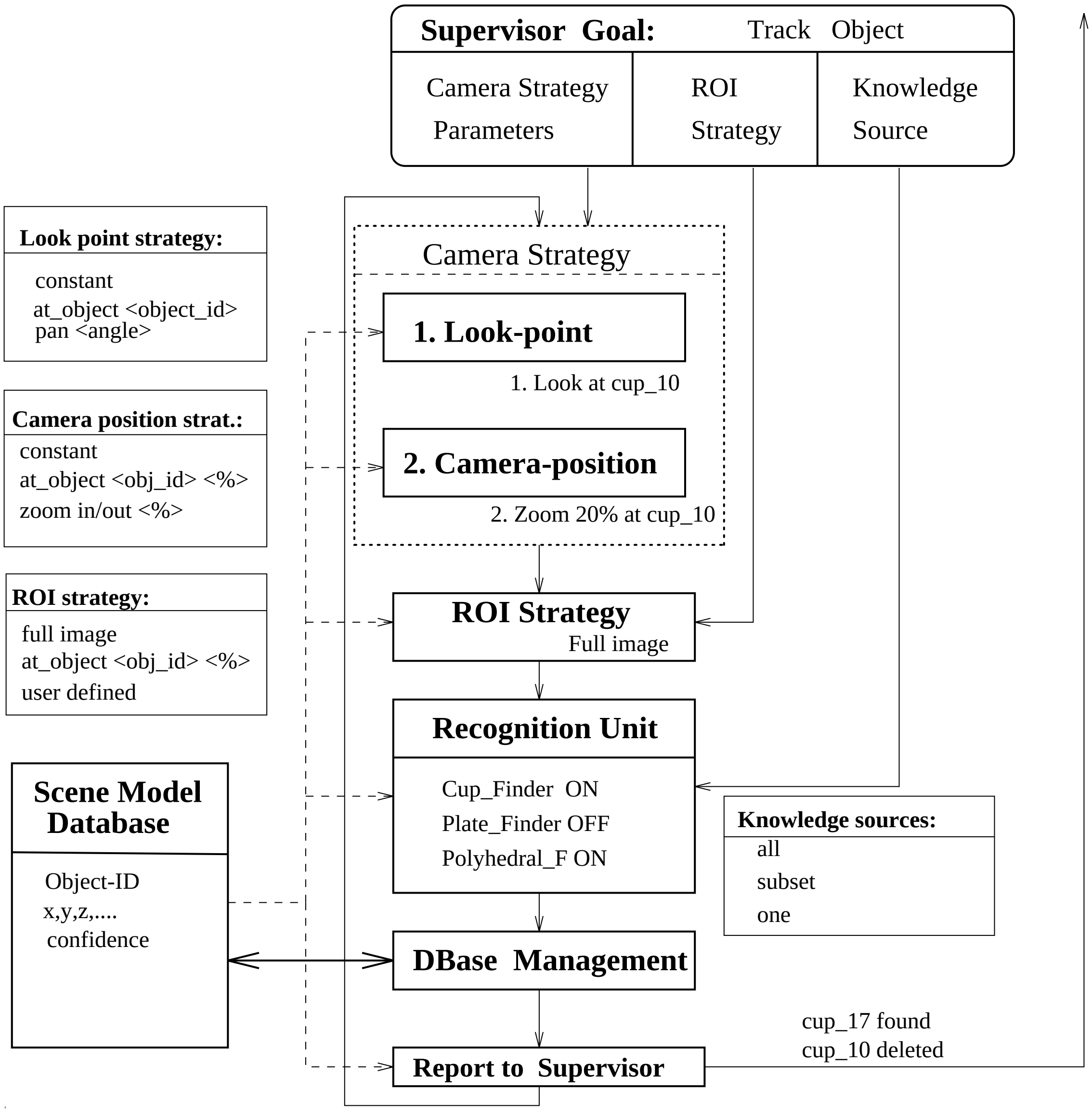

Figure 2: Interpretation module architecture

database serves. The pose and spatial extent attributes (modelled by a minimal bounding parallelopiped) form a minimum object description sufficient for geometric reasoning used for verification of basic physical constraints (eg. no two objects occupy the same space). The pose and extent attributes are passed to the camera strategy unit when sensor attention is centred at a known object. The class information is related to the organisation of the recognition unit. All available recognition knowledge sources form a hierarchical tree (the following sequence can serve as an example of a branch: rotationally symmetric objects - cylindrical objects - cylindrical objects with similar diameter and hight). If a verification or update of parameters for a particular object is requested by the supervisor then the class attribute is used to invoke the appropriate (least general) recognition knowledge source; in conjunction with a unique object ID this enables fast indexing and retrieval of a detailed, object specific model with instantiated parameters, and, consequently, a more efficient and robust results are obtained.

It is important to stress that all objects, regardless of the recognition class, are represented in an identical way. All units within the module 'understand' the generic object representation and can access or store the information in a uniform way. Moreover, further refinement of the recognition class hierarchy or an introduction of a new knowledge source require changes only in the recognition unit as all class specific information is managed by the appropriate recognition knowledge source.

The building of a symbolic scene model in a system with an active mobile sensor is possible only when a coordinate transformation between the scene model reference coordinate systems and the camera coordinate system can be established and maintained. In order to avoid the accumulation and amplification of errors a hierarchy of local reference coordinate systems is used. The registration of the camera coordinate system and a local reference frame is established by means of recognition of an object whose internal coordinate system defines the reference frame.

The environment in which the interpretation module operates is constantly changing. Information about the pose of moving objects is out of date even before it is inserted in the scene model. The database manager takes this fact into consideration when updating confidence values of object attributes. When no new evidence about an object is available, eg. when the object is not in the field of view, the gradual aging of the pose estimate is modelled by an exponential decrease of the confidence level; the object mobility (durability) attribute defines the decay constant and hence the speed of the exponential forgetting process. For objects in the field of view, the evidence for object existence is temporally integrated using a simple counting scheme that updates confidence levels. Initially, an object hypothesis is assigned a confidence value equal to the likelihood of a correct match (the likelihood is part of the information output by the recognition unit). If a corresponding object is found in subsequent frames the confidence level in the hypothesis is increased (proportionally to the match likelihood) until a maximum value indicating absolute belief in the presence of an object is reached. Two hypotheses are assumed to correspond to a single object if the pose and extent parameters suggest volumetric overlap. Non-observation is taken as negative evidence and the confidence in the presence of the object is decreased by a constant. A velocity model for the motion and tracking of moving objects is not explicitly included in the current implementation.

3.2 Camera strategy unit

Goal driven sensor control is an indispensable part of an active vision system. Unlike a conventional camera controller, the camera strategy unit exploits the information stored in the scene module. Thus, instead of specifying the desired camera position and look point in terms of n-tuples of (Cartesian) coordinates, the supervisor goal defines camera movement indirectly by reference to objects stored in the database; eg. 'look_at cup_17 position 70%' (meaning: rotate the camera so that the center of cup_17 lies on the optical axis, select a viewpoint so that the cup projection occupies 70% of the field of view). This camera strategy abstraction simplifies the mapping between a high-level, user-defined goal (represented in a form close to natural language) from the mechanics of sensor movement. Moreover, increased modularity is achieved by insulating the supervisor from details of the database organization. The determination of camera parameters is not just a simple traversal of the reference frame tree and computation of camera-to-object coordinate system transformation; errors in object positions (the object of interest and those defining relevant local reference frames) must be taken into account (see [6] and [7] for details).

The following minimum capabilities are required of the underlying sensor controller effecting the camera strategy:

- (Cartesian) camera position control
- independent camera direction (look point) control

Other camera parameters, eg. focus, aperture, vergence (for a stereo pair), are assumed to be adjusted automatically by low level, reflexive as opposed to goal-driven, purposive processes. Speaking in terms of an analogy to the human body the interpretation module attempts to obtain a suitable viewpoint by controlling the head and neck rather than the eye movements.

The camera unit comprises two separate components, the look point strategy and the camera position strategy; table 1 summarizes their operational modes. The mapping of a supervisor goal into the operational modes is discussed in Section 3.5.

3.3 Region-of-Interest Unit

Selection of a restricted subset of incoming data for low-level processing, the main task of the ROI unit, provides a focus-of-attention mechanism complementary to active camera movements. The operational modes of the unit that determine the strategy applied for limiting the processed region, are listed in table 2. An interesting example of a symbolically defined ROI interest strategy, ie. using a reference to an object in the scene model, is demonstrated in the second experimental run described in section 5.

Besides the expected beneficial impact on processing speed and data complexity, the use of ROI proved to improve the accuracy and reliability of low level vision processes that base the setting of control parameters on automatic estimation of image noise level.

command	parameters	description
Camera position strategy		
constant	$\emptyset$	No change in camera position
at_object	object ID, %	Move the camera so that the object fills a <percentage> of the field of view.
zoom_in/out	%	Move forward/backward along the line of sight (ie. look direction) stretching/shrinking the current field of view according to the specified percentage
user_defined	x, y, z	Move the camera to a specified position
Look point strategy		
constant	$\emptyset$	No change in look direction
at_object	object ID	Keep the look point at the center of the specified object.
pan	angle	Rotate the camera in the horizontal plane by <angle> degrees.
user_defined	x, y, z	Direct the camera towards the specified point

Table 1: Operational modes of the camera strategy unit.

Region-of-interest strategy		
command	parameters	description
full	$\emptyset$	Process the full image(s)
at_object	object ID, %	Frame the 2D projection of the object bounding box, stretch/shrink the frame by '%' percent, pass the 2D limits to the low-level modules
user_defined	xmin, ymin, xmax, ymax	Process the specified area only

Table 2: Operational modes of the Region-of-Interest Unit.

3.4 The Recognition Unit

Bootstrapping operations of the interpretation module as well as supervisor goal satisfaction at least initially rely completely on the successful bottom-up performance of the recognition knowledge sources. The recognition unit serves the following purposes:

- provide an interface between the recognition knowledge sources and the scene model database, ie. pass information about
  - matched objects to the scene model manager
  - instantiated parameters of object hypotheses to individual knowledge sources to enable rapid indexing of relevant internal models
  - provide synchronisation for the recognition processes running in parallel.
- select and launch recognition KSs according to the types of objects which are the subject of the supervisor query

Currently three general KSs are running in the test environment; a polyhedral object matcher, a cylinder (cup) finder, and a plate finder (detector of rotationally symmetric planar objects). A detailed discussion of the knowledge sources is beyond the scope of this paper; see [12], [15], and [14] for description. The KSs draw on results of a complex set of low and intermediate level developments:

- a novel generalized Hough transform algorithm [9]
- a geometric modelling package [10]
- perceptual grouping algorithms providing an intermediate image description in terms of collinear and parallel lines and various types of junctions in 2D [8], [4], [5]
- a robust polygon extraction method [11], [13]

3.5 Visual Behaviour

As briefly indicated in section 2 the interpretation module operation is modified in response to goals specified by the system supervisor. The following set of user goals has been defined in the early stages of the VAP project [2]:

- Search: Determine if a particular object (class) is present in the scene.
- Find: Re-find an object which has been recognised earlier.
- Watch: Allocate resources for maintenance of the description of a particular object.
- Track: Maintain description of a particular object and report continuously to the user.
- Explore: perform bottom-up driven exploration of the scene.

Look point	Camera position	Region of Interest	Recognition Knowledge Sources	Description
constant	constant	full image	all	**explore** a static, predefined area
constant	constant	full image	some	**watch** for a certain type of object(s) in a predefined area (ie. selective explore)
constant	constant	full image	one	**search** for an object of a specific type in a predefined area
pan $<angle>$	constant	full image	all, some, one	• wide area **explore/watch/search**; the camera is panning $<angle>$ degrees after every frame
constant	zoom in/out $<\%>$	full image	all, some, one	• **explore/watch/search** with zoom-in or zoom-out (ie. camera moves along the line of sight)
constant	constant	user defined	all, some, one	**explore/watch/search** a user-defined area of the image

Table 3: Primitive behaviours applicable in the initialisation (bootstrap) phase, ie. before any particular objects are recognised. Behaviours marked with a '•' require a movable camera head.

Look point	Camera position	Region of Interest	KS	Description
at_object <object_id>	constant	full image	one	**look at** an object. If the object moves, it is automatically **tracked** as the look point controller tries to keep it in the center of the field of view.
at_object <object_id>	at_object <object_id> <%>	full image	one	•**zoom on** an object. Keep the camera look point at the center of the object bounding box. Camera position ensures that the object covers <%> of the image area. To achieve the 'constant' object projection the camera must follow the object.
at_object <object_id>	constant	at_object <object_id> <%>	one	•**focus on** an object. Keep the camera look point at the center of the object bounding box. Process only a selected region of interest around the object. Note that this mechanism is similar to **zoom on** - the object of interest fills a constant portion of the processed part of the image. Here, efficiency is gained as smaller part of the image is processed; **zoom on** effectively increases resolution.
pan <angle>	at_object <object_id> <%>	full image	one	•**multiple views** of an object. As the look point mechanism pans the camera and the camera positioning module keeps the object in the center of the field of view, image of a single object from different views are acquired.
constant	constant	at_object <object_id> <%>	one	**dynamic region of interest**. The region of interest follows a moving object. Can be used for simple experiments with focus of attention without a movable camera head.

Table 4: Primitive behaviours implementing various forms of focus of attention. Behaviours marked with a '•' require a movable camera head.

To achieve external goals, the supervisor performs detailed planning that results in a sequence of goals defining the operational modes of the functional units of the interpreter. From the point of view of an external observer, the operational mode of the interpretation module manifests itself as a primitive visual behaviour (the word 'primitive' is used to distinguish between the supervisor response to a user goal, called visual behaviour, and its components).

The operational modes of the camera strategy, region-of-interest and recognition units has been presented in previous subsections (3.2,3.3, 3.4). Originally, the operational modes, and indeed the whole structure of the interpretation module, were designed so as to facilitate satisfaction of the set of user goals listed in the beginning of the section. Then, in an attempt to justify the proposed module structure, the inverse problem was investigated: 'would all combinations of operational modes, ie. all primitive behaviours, result in a sensible, intuitively compelling behaviour?'. In fact,a number of previously unforeseen, reasonable primitive behaviours were discovered; see tables 3.5 and 3.5 for details. The tables list just the most compelling primitive behaviours as the number of possible combinations of operational modes is large. The definition of primitive behaviour in terms of modes of operation allowed for transformation of the ad-hoc group of supervisor goals into a much wider and yet consistent set (no combination of operational modes results in a senseless behaviour).

3.6 Loop of Perception

The complexity of the control strategy necessary for efficient operation of the interpretation module is greatly reduced by two factors. First, all planning for user goal satisfaction is performed at the supervisor level. Second, the resulting sequence of supervisor goal commands directly modifies the operational modes of the camera strategy, region-of-interest and recognition units. In this distributed control framework the module controller's main responsibility is to synchronise the cooperation of the individual independent units effecting the required *perceptual behaviour*. As perceptual behaviours are structurally identical, the interpretation process can be accomplished within a continuous, fixed cycle of operation - *the loop of perception*. The following four stages are repeated in the loop (see fig. 2):

1. Operational modes of all units are set according to the current supervisor goal.

2. Initiate the appropriate processes to determine the camera next look direction and position, and the region of interest in the image, using temporal context as required.

3. The relevant recognition knowledge sources are enabled and their launching is triggered by the low level image description as soon as it becomes available. The output of the knowledge source(s) is stored in the symbolic scene model database together with any confidence factors computed during the matching process.

4. The scene interpretation module controller reports the status of goal achievement to the supervisor.

General database management (confidence updates, garbage collection etc.) is performed in parallel within all phases. Data are passed between units indirectly through the scene model as described in section 3.1. The phases of the loop of perception are not stages of the execution flow; they should be rather viewed as sequence points when a certain set of parallel operations, necessary for the following phase, is completed.

4 Implementation Notes

The core of the interpretation module - the controller, camera strategy unit, region of interest unit and the recognition units are written in a public-domain production language CLIPS version 4.2 [16]. CLIPS, "... a type of computer language designed for writing applications called Expert Systems"([16], p. 1) facilitated fast prototyping and module implementation without compromising on efficiency (which was not of great concern anyway as the interpretation process spends most of the time waiting for the low-level image processing to be completed) . Small, specialized parts of the code were written in C (matrix and vector packages, transformations between coordinate frames). Interfacing C and CLIPS is seamless (CLIPS stands for "C" Language Integrated Production System).

The interpretation module can run as a part of the VAP system or in a stand-alone mode. In a stand-alone mode, useful especially for debugging and testing, communication with external modules (camera head, supervisor) is implemented with the help of control files. Within the VAP system, communication facilities are provided by the system skeleton SAVA.

5 Experiments in High level vision

The interpretation module presented in previous sections has successfully processed several image sequences. During the final review of the VAP I project, the module was in operation for several hours, interpreting scenes of breakfast scenario type, containing a table, plates, cups, and boxes. The objects in the scene were moved around as one would expect in such a scenario.

A sequence of images (figures 4-9) will be used to demonstrate the operation of the module. The objects involved are: 2 cups (one moving), a box, a plate (moving) and a table. The first experiment (figures 4-6) focuses on operation of individual recognition knowledge sources and on scene model database maintenance. In the second experiment, run on the same sequence, temporal context is exploited using the region-of-interest mechanism. Behaviours with active camera control are not presented in this paper; experiments including camera motion were only in preliminary stages at the time of writing of the report.

It is assumed in the experiments that object 'table' has already been identified; the tabletop plane to camera transformation is therefore known. The previous stage of interpretation concerned with establishing table-to-world coordinate transformation, a particular case of a reference frame tree transversal, is described elsewhere [7].

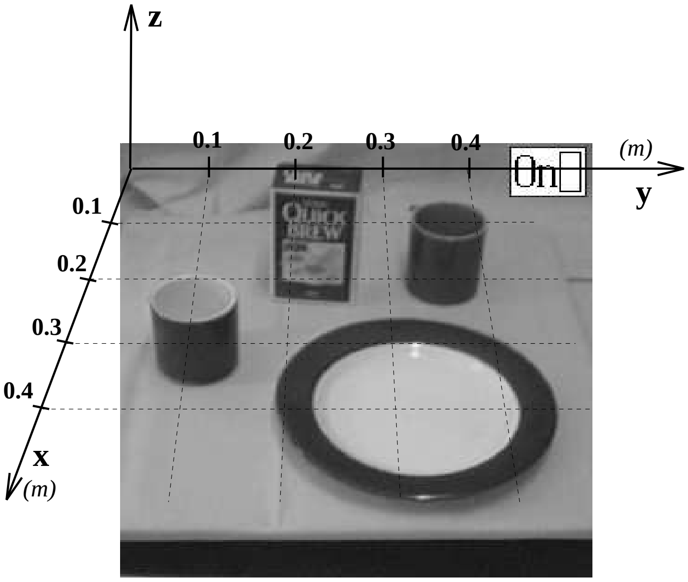

Figure 3: The 3D coordinate system, the table-top reference frame, used in the two experiments

The two experiments will be presented as a commented log output by the system. Information about the interpretation process is formatted as in the following template:

```
----------- Frame frame_number ----------
perceptual behaviour: supervisor goal
  look point        : command (mode of operation)
  camera position   : command (mode of operation)
  region of interest: command (mode of operation)
  knowledge sources : list of knowledges sources

+ object-type_ID [confidence_level] 3D_position
+ object-type_ID [confidence_level] 3D_position
  object-type_ID [confidence_level] 3D_position
- object-type_ID [confidence_level] 3D_position
```

The log entry starts with the processed 'frame number'. The 'frame number' should help readers to relate the numerical information about object position presented in the log file to the graphical results shown in figures 4-9. Next, the supervisor goal defining module's perceptual behaviour is displayed. To save space, this information is shown only when a new goal is received. The operational modes of the look point, camera position, region of interest, and knowledge source units follow. The complete list of commands (operational modes) can be found in tables 1 and 2. The last section of the log entry contains information about object stored in the scene database. Object related data is structured as follows:

1. • '+' marks newly detected objects

- '−' marks objects to be removed from the scene model

2. object_type and a unique ID
3. confidence level of object hypotheses [in square brackets]
4. x,y and z triplet defining the 3D position of the object centroids. The 3D coordinate frame used in the experiment, the local reference frame of the tabletop, is depicted in figure 3.

Unfortunately, information about time was not included in the log. It is difficult to give general estimates of the time needed to complete one cycle of perception as numerous factors, mainly outside the interpretation module itself, determine the overall performance. One of the most significant factors, the execution time of low-level image processing, can be eliminating by use of special-purpose hardware developed within the VAP project. The timing is also significantly influenced by the speed and configuration of the local network as units of the interpretation module (region-of-interest, camera control, database management) as well as individual recognition knowledge sources (plate, cup and polyhedral object finders) run in parallel ; (the system has been tested on a number of heterogeneous networks of UNIX machines comprising Sun3s, Sun4s, SPARC Is and IIs). Last but not least, the execution time is greatly reduced by applying any of the focus-of-attention strategies. In the worst case scenario (all KS running, all image processed, no special purpose HW, a single SPARC 2) the time of one cycle is roughly 1min; on average (multiple SPARC IIs, no image processing HW, region of interest focused on a object) a loop of perception is completed in less then 5 seconds .

In the following sections two experiments are presented.

5.1 Experiment 1: Explore

This section discusses an experiment which demonstrates operation of the interpretation module in the data-driven, *explore* mode.

```
----------- Frame 1 ----------
perceptual behaviour: explore
  look point        : constant
  camera position   : constant
  region of interest: full image
  knowledge sources : plate_finder cup_finder poly_finder

+ poly_1  [1.00] 0.01 0.24 -0.07
+ plate_9 [1.00] 0.42 0.29 0.00
+ cup_11  [1.00] 0.34 0.12 0.04
```

The scene model database is empty in the beginning of the interpretation process; so far no objects have been recognised. The supervisor can therefore issue only one of the bootstrapping goals listed in table 3.5 (ie. a goal not requiring object ID as a parameter). The log listing shows that the interpretation process is operating in the *explore* mode. The selection of the *explore* behaviour need not be a consequence of receiving a supervisor goal; the interpretation module enters the default *explore* mode automatically if no supervisor goal is issued (all other changes in perceptual behaviour are induced by supervisor goals).

The perceptual goal description confirm that all available knowledge sources are launched (plate, cup and polyhedral object finder). The recognition knowledge sources detected three objects labelled poly_1, plate_9 and cup_11.

Comparison of the database contents and image 1a of figure 4 indicates that the estimate of plate_9 and cup_11 pose is good. The pose of tea box poly_1 is estimated at 7cm below the tabletop plane - an obvious error. Moreover, the dark cup in the upper-left corner was not found. The imperfect performance of the recognition procedures is not of great concern; our approach is based on the assumption that temporal integration of (inherently noisy) recognition results is robust enough to lead to stable scene interpretation.

```
----------- Frame 2  ----------
- poly_1  [0.00] 0.01 0.24 -0.07
  plate_9 [3.00] 0.42 0.29 0.00
  cup_11  [2.00] 0.34 0.12 0.04

----------- Frame 3 ----------

  plate_9 [4.00] 0.42 0.29 0.00
  cup_11  [3.00] 0.35 0.12 0.04
+ poly_22 [1.00] 0.37 0.24 0.14
```

The presence of plate_9 and cup_11 in the scene has been confirmed by recognition knowledge sources in frames 2 and 3. Confidence levels for both objects have been increased using the updating scene discussed in section 3.1. The weak poly_1 hypothesis was removed when its confidence level dropped to 0. A new (and once again incorrect) polyhedral object hypothesis was instantiated in frame 3.

```
----------- Frame 4 ----------

  plate_9 [5.00] 0.42 0.29 0.00
  cup_11  [4.00] 0.36 0.11 0.04
- poly_22 [0.00] 0.37 0.24 0.14

----------- Frame 5 ----------

  plate_9 [5.00] 0.42 0.29 0.00
  cup_11  [4.50] 0.39 0.10 0.04
+ poly_40 [1.00] 0.31 0.22 0.13
```

In frame 4 the confidence level of the plate_9 hypothesis has reached the maximum, 'absolute certainty' level. The increase in confidence in the cup_11 hypothesis is smaller as the cup, contrary to the stationary plate, moves. The change in the cup position forces the confidence updating scheme to consider various options, eg. 'Am I getting very noisy measurements of a static cup pose?', 'Is the cup moving?', 'Perhaps the old cup was taken away and a new one put close to the original cup?'. Although the right interpretation (ie. motion) is chosen, the confidence level is updated with more restraint to cater for the other options.

Note the good agreement between the cup trajectory in images 1-5 of figure 4 and the recorded data. The poly_22 hypothesis was removed from the database; the new box hypothesis, poly_40, is very close to reality.

```
----------- Frame 6 ----------
perceptual behaviour: watch (cup, plate)
  look point        : constant
  camera position   : constant
  region of interest: full image
  knowledge sources : plate_finder cup_finder


  plate_9 [5.00] 0.42 0.29 0.00
  cup_11  [5.00] 0.43 0.12 0.04
  poly_40 [0.98] 0.31 0.22 0.13
```

In frame 6 a new supervisor goal, *watch (cup,plate)*, is issued. The operational modes of the basic units are modified accordingly. In this case only the polyhedral knowledge source is turned of; the camera and region of interest strategy remains unchanged. The new supervisor goal could be issued as a consequence of :

- a user request
- detection of a triggering event in the data passed to the supervisor
- entering a new stage of a dynamic plan

However, reasoning about perceptual behaviour is not in the scope of the interpretation module.

```
----------- Frame 7 ----------

  plate_9 [5.00] 0.42 0.29 0.00
  cup_11  [5.00] 0.46 0.12 0.04
  poly_40 [0.96] 0.31 0.22 0.13

----------- Frame 8 ----------

  plate_9 [5.00] 0.42 0.29 0.00
```

```
  cup_11  [5.00] 0.48 0.12 0.04
  poly_40 [0.94] 0.31 0.22 0.13

----------- Frame 9  ----------

  plate_9 [5.00] 0.42 0.29 0.00
  cup_11  [5.00] 0.48 0.12 0.04
  poly_40 [0.92] 0.31 0.22 0.13

----------- Frame 10 ----------

  plate_9 [5.00] 0.41 0.28 0.00
  cup_11  [3.00] 0.48 0.12 0.04
  poly_40 [0.90] 0.31 0.22 0.13
```

No new objects are detected in frames 7-10. The pose of object plate_9 is very stable, changing within the system precision of 1cm. Images 7a-9a show that movement of cup_11 was correctly followed. At image 10 the cup was removed from the scene and consequently the cup recognition source didn't detect it. The confidence level was decreased, but the object hypothesis will remain in the scene model for another 2-3 frames until the confidence falls to 0.

The confidence level of the poly_40 hypothesis decreases slowly from frame 6 when the polyhedral knowledge source was switched off. The updates of confidence in poly_40 and cup_11 model two completely different phenomena. In the case of cup_11 strong evidence is produced by the recognition knowledge source that the object has disappeared. No such information is available about poly_40; the module is effectively 'blind' to polyhedral objects as the appropriate knowledge source is not running. However, the scene model must take into account the *aging* of the information related to poly_40. The speed of confidence change is governed by the expected average durability of object pose of an 'unviewed' tea-box (the same strategy is used for objects not in the field of view).

```
----------- Frame 11 ----------

  plate_9 [5.00] 0.40 0.27 0.00
  cup_11  [2.00] 0.48 0.12 0.04
  poly_40 [0.88] 0.31 0.22 0.13

----------- Frame 12 ----------

  plate_9 [5.00] 0.38 0.25 0.00
  cup_11  [1.00] 0.48 0.12 0.04
  poly_40 [0.86] 0.31 0.22 0.13
```

```
----------- Frame 13 ----------

  plate_9 [5.00] 0.38 0.23 0.00
- cup_11  [0.00] 0.48 0.12 0.04
  poly_40 [0.84] 0.31 0.22 0.13
```

The interpretation process progresses as expected in frames 10-13. The cup_11 hypothesis is finally removed. The left-to-right motion of plate_9 is correctly tracked.

```
----------- Frame 14 ----------

  plate_9 [5.00] 0.37 0.21 0.00
  poly_40 [0.82] 0.31 0.22 0.13

----------- Frame 15 ----------
perceptual behaviour: explore
  look point        : constant
  camera position   : constant
  region of interest: full image
  knowledge sources : plate_finder cup_finder poly_finder

  plate_9 [5.00] 0.38 0.20 0.00
  poly_40 [0.80] 0.31 0.22 0.13
+ cup_116 [1.00] 0.24 0.35 0.04
```

The supervisor switches the mode back to *explore*. A new cup hypothesis is created.

```
----------- Frame 16 ----------
  plate_9  [5.00] 0.38 0.19 0.00
  poly_117 [1.00] 0.11 0.23 0.01
-  cup_116 [0.00] 0.24 0.35 0.04
```

In the last frame a new polyhedral object, poly_117, is detected. It is not recognised that it is actually a noise measurement induced by the same tea-box as poly_40; the hypothesis are not merged, the old is replaced by the new one.

5.2 Experiment 2: Focus of Attention

The first experiment presented the basic functionalities of the scene interpretation module: the temporal integration of recognition results, confidence maintenance, change of perceptual behaviour in response to a supervisor goal. In the second experiment an additional mechanism, region of interest control, is used to implement focus of attention.

```
----------- Frame 1 ----------
perceptual behaviour: explore
  look point         : constant
  camera position    : constant
  region of interest: full image
  knowledge sources : plate_finder cup_finder poly_finder


+ poly_1 [1.00] 0.01 0.24 -0.07
+ plate_9 [1.00] 0.42 0.29 0.00
+ cup_11  [1.00] 0.34 0.12 0.04
```

The focus-of-attention and explore experiments process the same sequence; consequently, the results will be identical as long as the perceptual behaviour (supervisor goal) remain the same.

```
----------- Frame 2  ----------

- poly_1 [0.00] 0.01 0.24 -0.07
  plate_9 [3.00] 0.42 0.29 0.00
  cup_11  [2.00] 0.34 0.12 0.04


----------- Frame 3 ----------
perceptual behaviour: track (cup_11 150)
  look point         : constant
  camera position    : constant
  region of interest: at_object cup_11 150%
  knowledge sources : plate_finder cup_finder

  plate_1 [2.98] 0.42 0.29 0.00
  cup_11  [3.00] 0.35 0.12 0.04
```

At frame 3 a new supervisor goal, *track (cup_11 150)* is issued. The region of interest unit operational mode is modified to ensure that only a restricted area around cup_11 is processed by the recognition knowledge sources. The 2D region-of-interest is computed as follows. First, all visible corners of the object's 3D bounding box are projected on the image plane. The minimum and maximum coordinate values in horizontal and vertical directions define the 2D bounding rectangle. The 2D region-of-interest (ROI) is obtained by stretching the bounding rectangle by a factor defined by the second parameter of the track command (clipping is performed if a part of the region-of-interest is outside the image). The above definition guarantees that the symbolic, object-centered ROI dynamically follows the object of interest.

The cooperation of the ROI mechanism with the recognition knowledge sources is carried out through information stored in the symbolic scene model. After every frame, the 3D bounding box information necessary for ROI computation is updated using the recognition results. The benefits of ROI are twofold: First, a significant speed up (approx. 10x in the presented experiment) is achieved as the recognition time is roughly proportional to the number of pixels processed. Second, low-level image processing routines that use image statistics for self-tuning (eg. automatic threshold selection for hysteresis linking in edge detection) improve performance because only information in the vicinity of the object is taken into consideration. In the presented experiment, only the second type of benefit can be observed; the speed up would normally result in more frequent acquisition of images (compared with the *explore* mode). Consequently, a larger number of images (per unit time) with smaller object movements would be processed. However, both experiment were performed on a pre-recorded sequence of 15 frames.

The 'stretch' parameter of the track command enables the supervisor to control the area of processing. The parameter setting, based on previous object mobility and loop of perception execution time, must reflect the following relationships. On the one hand, a smaller area of processing means faster execution of the loop of perception while on the other hand the object of interest can (partially) move out of a small ROI and thus render detection impossible.

```
----------- Frame 4 ----------

  plate_9 [2.96] 0.42 0.29 0.00
  cup_11  [3.50] 0.38 0.10 0.04

----------- Frame 5 ----------

  plate_9 [2.94] 0.42 0.29 0.00
  cup_11  [4.50] 0.38 0.10 0.04

----------- Frame 6 ----------

  plate_9 [2.92] 0.42 0.29 0.00
  cup_11  [5.00] 0.42 0.12 0.04

----------- Frame 7 ----------

  plate_9 [2.90] 0.42 0.29 0.00
  cup_11  [5.00] 0.46 0.12 0.04
```

The cup_11 object is successfully tracked in frames 4, 5, 6 and 7. Comparison of the cup_11 pose with results obtained in the *explore* experiment shows that pose estimates in the two sequences are close (within 2cm) but not identical. This observation can be explained by adaptation of the low-level modules (edge detection) to noise specific to the ROI. The plate_9 object is effectively out of the field of view; its confidence level is updated according to the forgetting scheme described in the presentation of the 'explore' experiment.

```
----------- Frame 8 ----------

  plate_9 [2.88] 0.42 0.29 0.00
  cup_11  [5.00] 0.46 0.12 0.04

----------- Frame 9  ----------

  plate_9 [2.86] 0.42 0.29 0.00
  cup_11  [4.00] 0.46 0.12 0.04
----------- Frame 10 ----------
perceptual behaviour: track (plate_9 150)
  look point        : constant
  camera position   : constant
  region of interest: at_object plate_9 150%
  knowledge sources : plate_finder cup_finder


  plate_9 [3.86] 0.41 0.28 0.00
  cup_11  [3.98] 0.46 0.12 0.04
```

Cup_11 disappears after frame 7. The supervisor responds to the decrease of confidence in cup_11 by issuing a new command *track plate_9 150*. The decrease of confidence in cup_11 proceeds according to the slow, out-of-field-of-view scheme as cup_11 is not fully inside the new ROI.

```
----------- Frame 11 ----------

  plate_9 [4.86] 0.40 0.27 0.00
  cup_11  [3.96] 0.46 0.12 0.04

----------- Frame 12 ----------

  plate_9 [5.00] 0.38 0.25 0.00
  cup_11  [3.94] 0.46 0.12 0.04

----------- Frame 13 ----------

  plate_9 [5.00] 0.38 0.23 0.00
  cup_11  [2.94] 0.46 0.12 0.04


----------- Frame 14 ----------
perceptual behaviour: watch (plate, cup)
```

```
  look point        : constant
  camera position   : constant
  region of interest: full image
  knowledge sources : plate_finder cup_finder


  plate_9 [5.00] 0.37 0.21 0.00
  cup_11  [1.94] 0.46 0.12 0.04
```

The processing proceeds as expected. The plate_9 was successfully tracked. Confidence in cup_11 starts to decrease rapidly when a new goal, *watch (plate, cup)* is received.

```
----------- Frame 15 ----------

  plate_9 [5.00] 0.38 0.20 0.00
  cup_11  [0.94] 0.46 0.12 0.04
+ cup_72 [1.00] 0.24 0.35 0.04

----------- Frame 16 ----------

  plate_9 [5.00] 0.38 0.19 0.00
- cup_11  [-0.06] 0.46 0.12 0.04
  cup_72 [0.00] 0.24 0.35 0.04
```

Due to the selected behaviour and original high confidence value the cup_11 hypothesis has not been discarded before frame 16 as no evidence against its presence was produced in the recognition process. However, if a more sophisticated geometric reasoning scheme was implemented, the cup hypothesis could have been disposed of when the volume of plate_9 intersected the space supposedly occupied by cup_11 thus creating an internal inconsistency in the scene model database.

5.3 Conclusion

A novel framework for control of scene interpretation has been proposed. It has be shown that decomposition of the interpreter into camera, region of interest and recognition units allows the module to respond to a broad class of visual goals by a simple mapping of the goal into operational modes of individual units. Moreover, the distributed architecture increases flexibility and maintainability of the interpretation module.

The two experiments described in section 5 clearly demonstrate the merits of the spatio-temporal context for scene interpretation. Information about objects in the scene is stored in a hierarchical scene model. Several 'forgetting' schemes are adopted to reflect dynamism of objects. The database is used to guide future spatially focus interpretation.

In the context of a prototypal indoor scene, the breakfast table-top, the interpretation module was capable to recognise, track and focus on objects with reasonable robustness and acceptable speed.

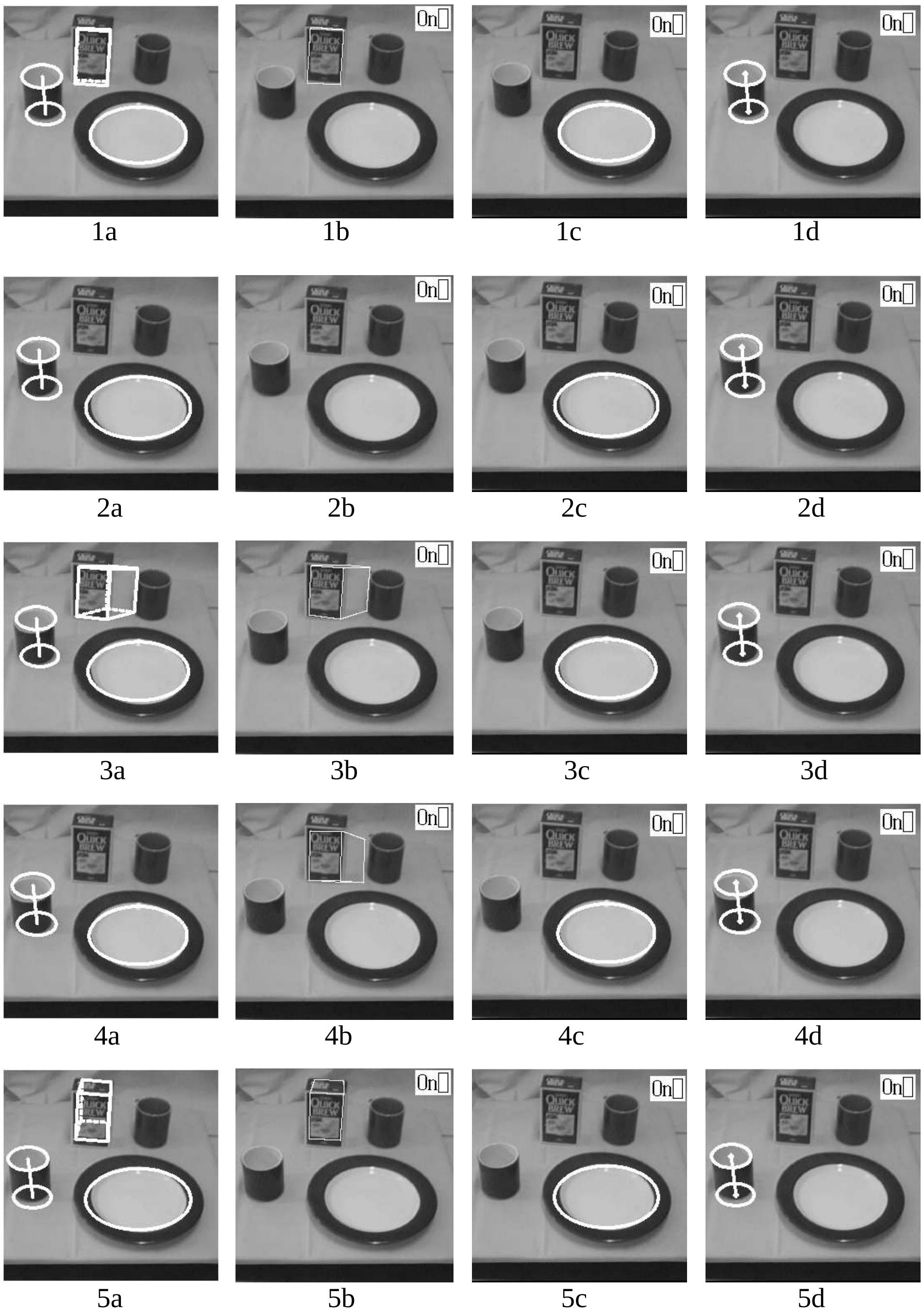

Figure 4: Experiment 1, frames 1-5. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

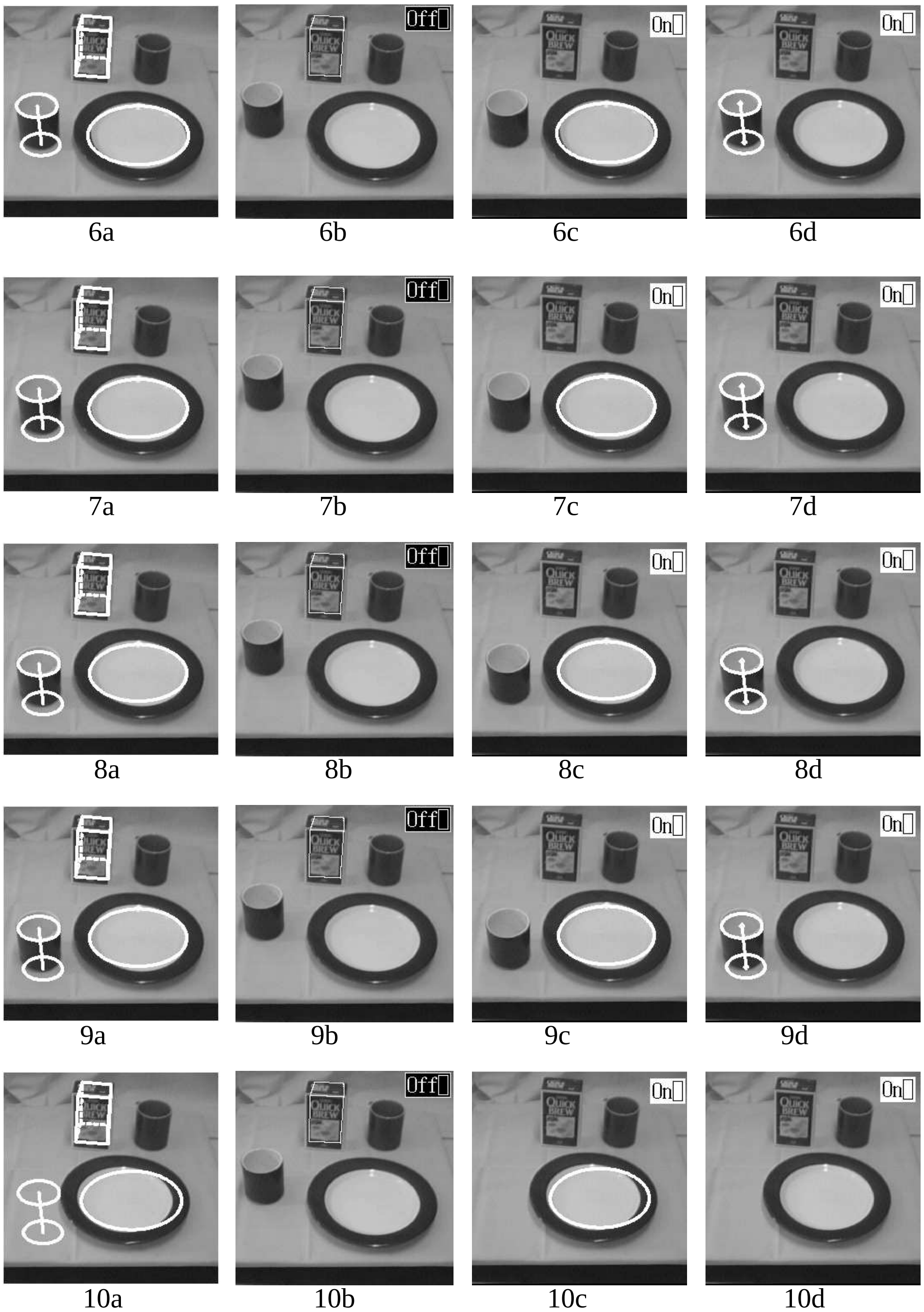

Figure 5: Experiment 1, frames 6-10. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

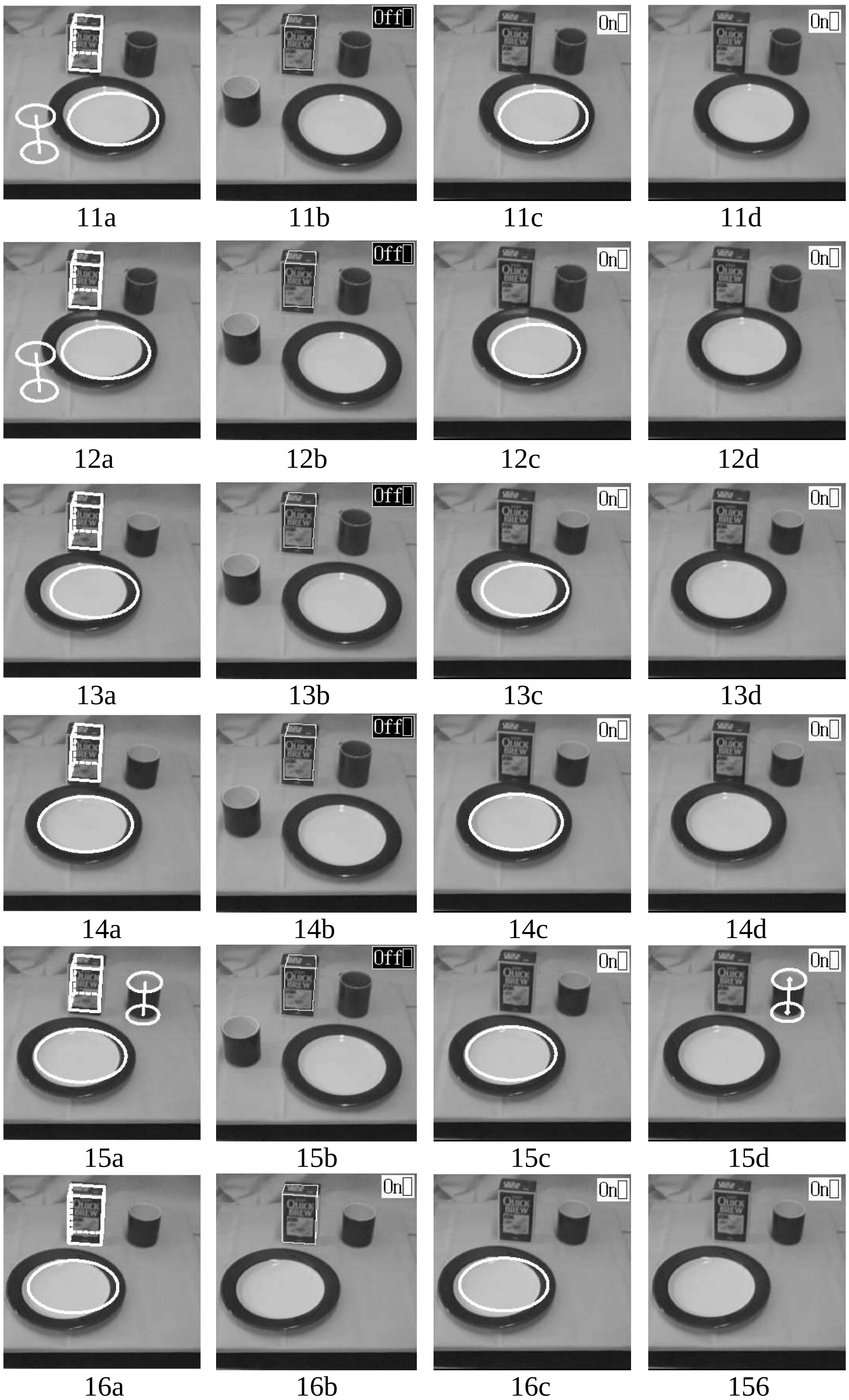

Figure 6: Experiment 1, frames 11-16. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

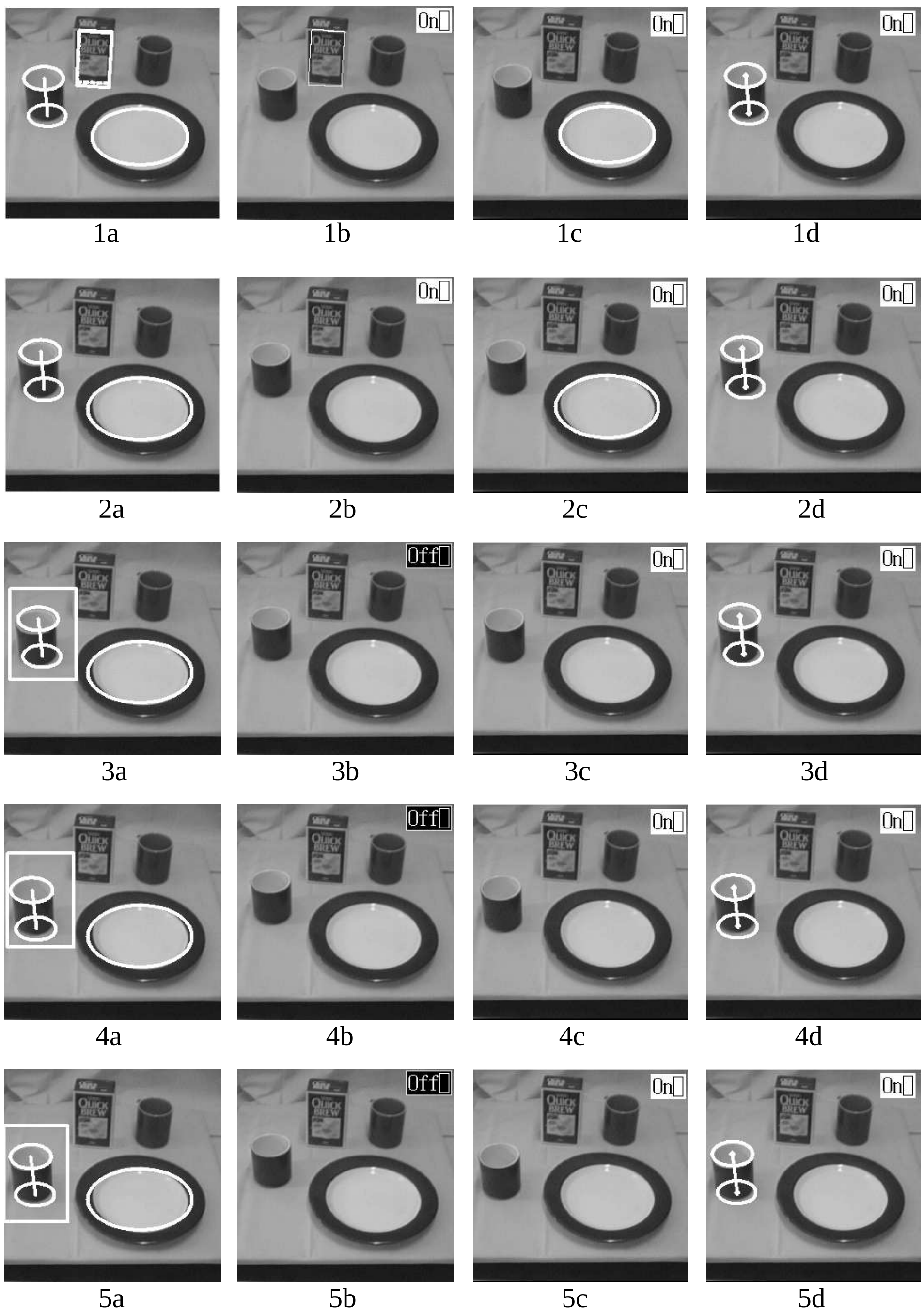

Figure 7: Experiment 2, frames 1-5. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

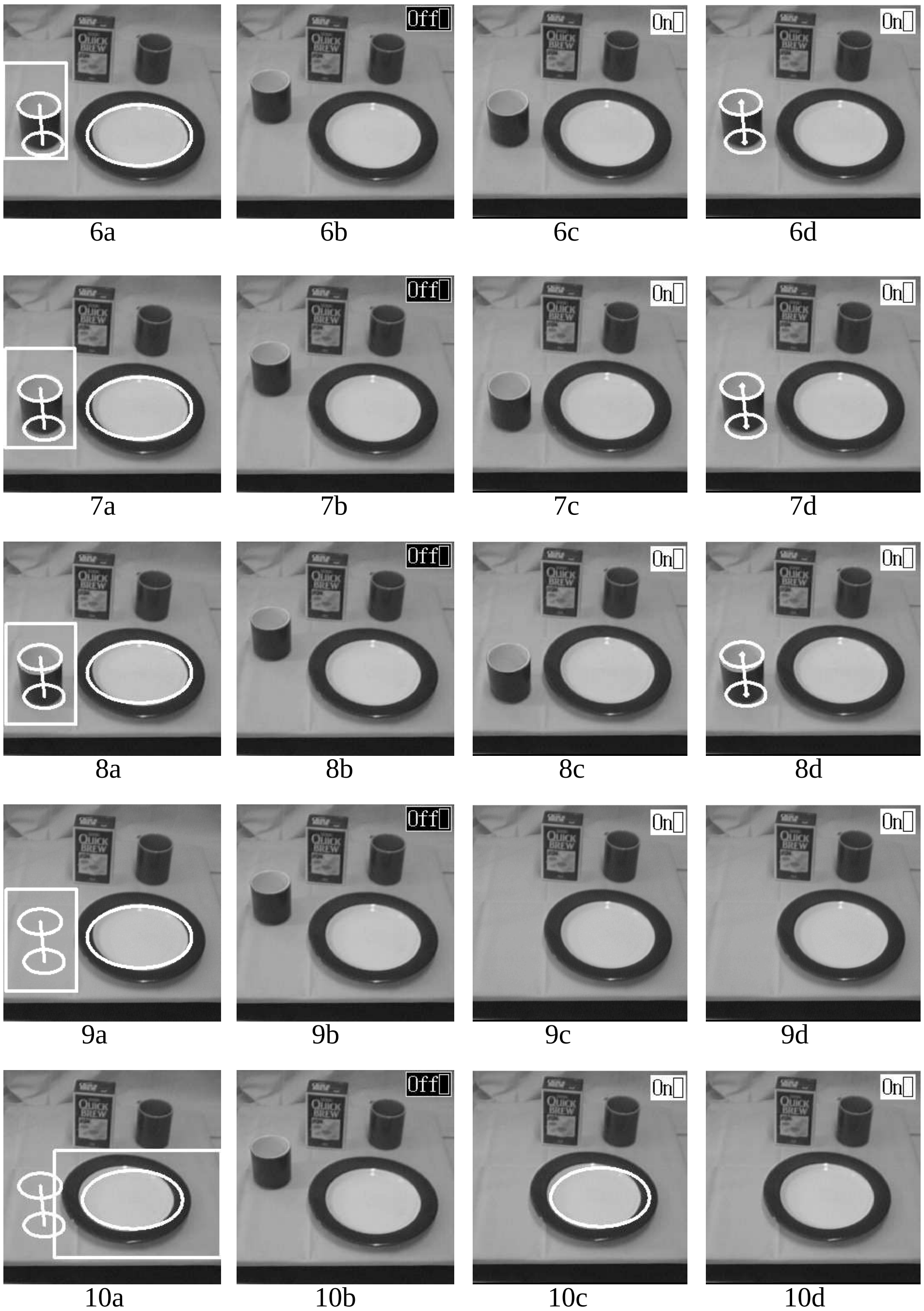

Figure 8: Experiment 2, frames 6-10. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

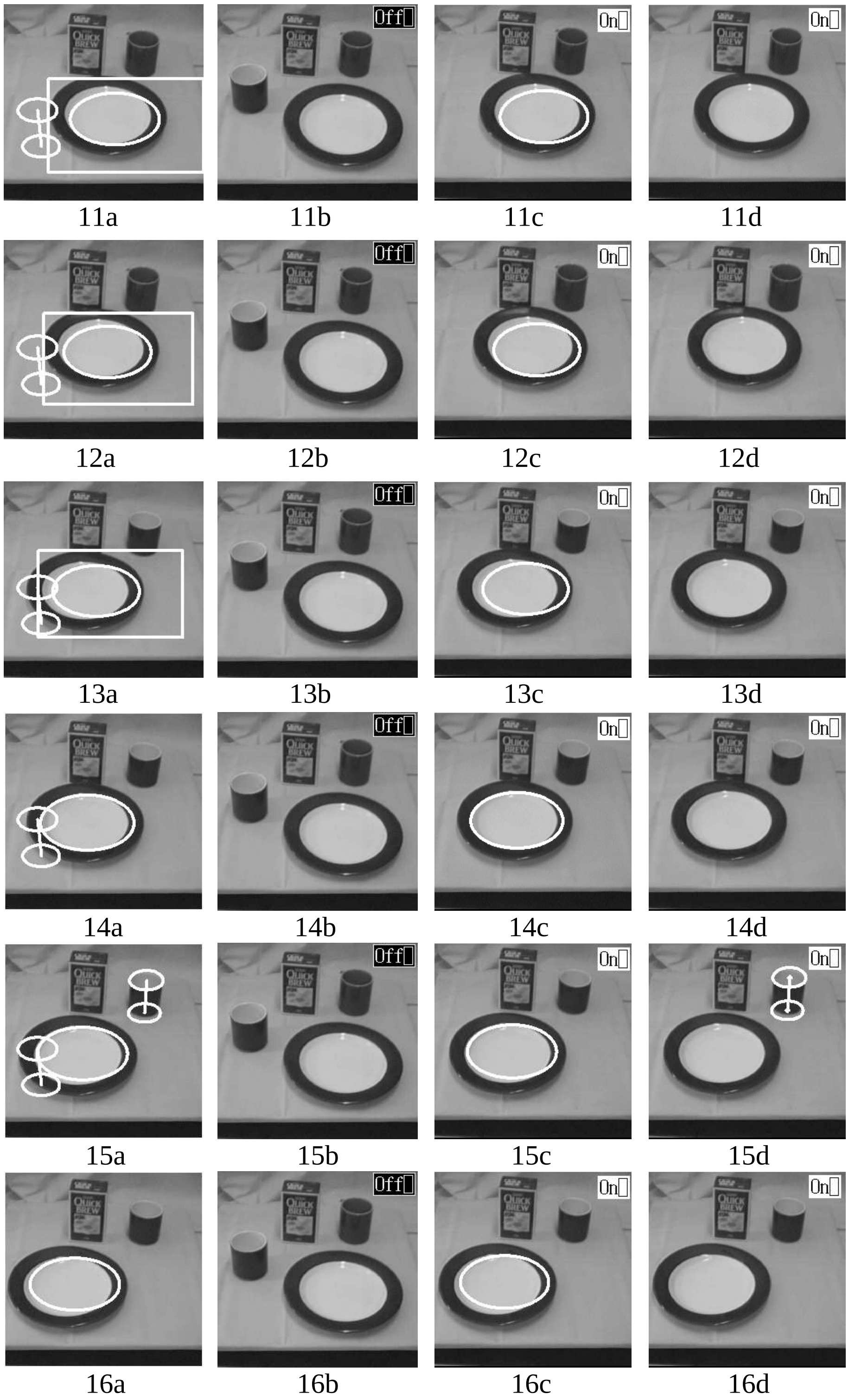

Figure 9: Experiment 2, frames 11-16. (a) projection of objects in the scene model into the image plane. (b)(c)(d) object hypotheses generated in the frame by polyhedral, plate, and cup recognition knowledge sources

References

[1] J. Kittler, J. Illingworth, G. Matas, P. Remagnino, K C Wong, H I Christensen, J.O. Eklundh, G. Olofsson, M.Li. Sybolic Scene Interpretation and Control of Perception VAP Project Document DD.G.1.3, March 1992.

[2] H I Christensen and E Granum: "Specification of Skeleton Control Structure", VAP Deliverable DR.E.1.2, May 1990.

[3] A Etemadi, J-P Schmidt, J Illingworth and J Kittler "Scene Description Representation", VAP Deliverable DR.D.2.3, August 1991.

[4] A. Etemadi, G. Matas, J. Illingworth, and J. Kittler. Low-level Grouping of Straight Line Segments. In *British Machine Vision Conference 1991*, Glasgow, September 1991. BMVC. pp. 118-126

[5] A Etemadi "Robust Segmentation of Edge Data", Proc 4th IEE Intern Conf Image Processing and Applications, Maastricht, 1992.

[6] P. Remagnino, G. Matas, J. Kittler, and J. Illingworth. On computing the next look camera parameters in active vision. In *10th European Conference on Artificial Intelligence*, Vienna, August 1992. ECAI. pp. 806-807

[7] P. Remagnino, G. Matas, J. Kittler, and J. Illingworth. Control in the bootstrap phase of a computer vision system. In *4th International Conference on Image Processing and its applications*, Maastricht, April 1992. IEE. pp. 85-88

[8] G. Matas, and J. Kittler. Contextual Junction Finder. In *British Machine Vision Conference 1992*, Leeds, September 1992. BMVC. pp. 119-128

[9] H M Lee, J Kittler and K C Wong"Generalised Hough Transform in Object Recognition", Proc 11 Intern Conference on Pattern Recognition, The Hague, 1992.

[10] K C Wong, J Kittler and J Illingworth "Analysis of Straight Homogeneous Generalized Cylinders Under Perspective Projection", Proc International Workshop on Visual Form, Capri 1991.

[11] K C Wong, J Kittler and J Illingworth:"Heuristically Guided Polygon Finding", Proc British Machine Vision Conference, pp 400-407, Glagow, 1991.

[12] K. C. Wong, Cheng Yu and J. Kittler, "Recognition of Polyhedral Objects Using Triangle-pair Features", The Special Issue of IEE Processings Part I on Image Processing, 1993. To appear.

[13] Y. Cheng, K. C. Wong and J Kittler"The Recognition of Triangle-pairs and Quadrilaterals from a Single Perspective View", Proc IEE 4th Internat. Conference on Image Processing and Its Applications, Maastricht, 1992.

[14] P. Hoad and J. Illingworth"Detection of Flat topped cylinders from a single view", submitted for publication

[15] P. Hoad"Extraction of Volumetric Primitives from 2D Image Data",MPhil/PhD transfer report, September 1992, Department of Electronic and Electrical Engineering, University of Surrey

[16] Joseph C. Giarratano" CLIPS User's Guide ", Artificial Intelligence Section, Lyndon B. Johnson Space Center, June 3, 1988

[17] Thomas D. Garvey, John D. Lowrance and Martin A. Fishler " An Inference Technique for Integrating knowldedge from Disparate Sources"

[18] Thomas D. Garvey"An Experiment with a System for Locating Objects in Multisensory Images", SRI International,

[19] Rodney A. Brooks "Visual Map Making for a Mobile Robot", In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision*, pp. 438-443

[20] Rodney A. Brooks " Symoblic reasoning among 3-dimensional and 2 dimensional images", *Artifcial Intellingence*, 17:349-385, 1983.

[21] A. R. Hanson and E M. Riseman " VISIONS: a computer system for interpreting scenes. In Allen R. Hanson and Edward M. Riseman, editors, *Computer Vision Systems.*, Academic Press, New York, 1978.

[22] Bruce A. Draper, Allen R. Hanson and Edward M. Riseman "Learning Knowledge-Directed Visual Strategies", In *Proc. of the DARPA Image Understandin Workshop*, 1992, pp. 933-940

[23] T. Lewitt, D. Lawton, D. Chelberg, P. Nelson and J. Due "Visual Memory for a mobile robot", In *Proc. of the AAAI Workshop on Spatial Reasoning and Multisensor Fusion, Morgan Kaufman Publishers, Los Altos, California, 1987*

[24] D.T. Lawton, T.S. Levitt, and P. Gelband " Knowledge based vision for terrestrial robots", In *Proc. of the DARPA Image Understandin Workshop*, 1988, pp. 933-940

[25] Raymond D. Rimey "Where to Look Next using a Bayesin Net: An Overview", In *Proc. of the DARPA Image Understandin Workshop*, 1988, pp. 927-932

[26] Thomas M. Strat and Grahame B. Smith "The Core Knowledge System", SRI International, Technical Note No. 426, October 1987

[27] Lambert E. Wixson "Real-time Qualitative Detection of Multicolored Objects", In *Proc. of the DARPA Image Understandin Workshop*, 1990, pp. 631-638

[28] Christopher M. Brown " Issues in Selective Perception ", In *Proc. of the DARPA Image Understandin Workshop*, 1992, pp. 21-30