Constraining Visual Expectations Using a Grammar of Scene Events

J Matas, J Kittler, J Illingworth, L Nguyen
Department of Electronic and Electrical Engineering,
University of Surrey, Guildford. GU2 5XH. United Kingdom.
and
H I Christensen
Laboratory for Image Analysis,
University of Aalborg, Aalborg. DK-9220. Denmark

Abstract

This paper considers the visual interpretation of dynamic scenes using temporal as well as geometric models. For this purpose scenes and events are represented via a grammar. Low level operators extract visual primitives which are matched to the statements of the grammar and a natural language description of significant scene events is produced. The further use of the grammar to constrain and predict visual events is considered and the method is illustrated on an breakfast setting domain.

1 Introduction

For many years the focus of computer vision research has been on the development of methodologies for solving discrete problems of scene recovery from image data. This bias, which is well documented in the computer vision literature e.g. [1, 2, 3], has been encouraged partly by the natural tendency of human beings to approach any complex problem by breaking it up into many small subproblems, and partly by the Marr's representational paradigm [16] which influenced the vision community in the nineteen eighties. According to his paradigm, single frames of image data are processed in a bottom up fashion so as to achieve scene reconstruction. The computational process corresponding to this paradigm involves launching discrete scene recovery algorithms (edge detection, shape from X, etc), the results of which are placed on a blackboard with the view to build a $2\frac{1}{2}D$ sketch of the imaged scene, as the basis of the eventual 3D interpretation.

In a recent departure from the Marr paradigm, [5, 4, 6, 8] argue that the complexity of solving vision problems can be greatly reduced by controlling both the sensor and visual data processing, focussing computational resources on a small part of the scene relevant to the specific goal. Although the efficiency of the *active vision* approach has been shown to facilitate real-time solution of real-world problems [10], an additional dimension to the complexity of an autonomous vision system is introduced by the problems of :

1. control and scheduling of knowledge sources providing different expertise regarding the visual data,
2. optimisation of knowledge source process parameters,
3. development of feedback strategies,
4. sensor control,
5. evaluation of the hypotheses generated by the knowledge sources,
6. scene model maintenance
7. visual goal definition

Recently these aspects of vision research have begun to attract the attention of the computer vision community. Some early work in the area of **control and scheduling of knowledge sources** is reported in [13, 12, 14, 9, 17]. The aim of the work was to determine an architecture, including means of communication, which would facilitate the operation of a complex vision system with multiple knowledge sources at all levels of processing.

The clear message from the research on algorithms for scene recovery is the need for continuous adaptation and tuning of their parameters to achieve the best possible performance. In research on topics in computer vision made in isolation, such optimisation is normally achieved manually for a set of images used for testing. It is apparent that such an approach is inappropriate in the case of autonomous vision systems. The first attempts at automatic **self optimisation of parameters** of vision system modules is reported in [19], but much more work is needed to develop a comprehensive solution to this problem.

While active vision raises practical problems of **sensor control** and **feedback strategies** it also gives rise to new conceptual problems. The most important of these is how to build and maintain a scene model that extends beyond the current field of view so as to provide an information data base and context to decide where to look next; what are the visual expectations, and how to define future visual goals. The past work addressing these issues is somewhat preliminary and focuses on static domains with either fully [11] or partially [7, 20] known environments, in which prior scene knowledge is encapsulated by specific or generic spatial object relations.

Solutions to the scientific problems mentioned above cannot be verified without an 'active' scene interpretation testbed. In [17] we describe the capabilities of the VAP Vision system; the complexity of the system (see fig. 1) poses a number of difficult engineering problems. In our previous work scene description in terms of recognised

objects with their 3D positions constituted the highest level of the external world model. The system could cope with a wide range of objects and visual phenomena such as motion patterns [20], but no explicit model of **scene evolution** existed.

In this paper we extend the scene description by adding the notion of a temporal and geometric **event**. Definition of geometric events is based on the non-accidentalness principle successfully used in object recognition [15]. If two objects assume a relative position that is unlikely to arise by accident it is inferred that the objects interact. The interaction constrains both object class and the 3D position. Temporal events concisely describe significant changes of object state. A grammar of temporal scene events ameliorates the vision system recognition performance by constraining visual expectations.

The work has been motivated by and builds on the extensive experience accumulated over the years in the area of traffic scene modelling and understanding e.g. [18] and references therein. Whereas the spatial model component in the traffic scene application is essentially 2D, our work aims to combine temporal models with full 3D spatial reasoning based on geometric events. As in [18], we propose to model temporal events linguistically. A grammar capturing possible sequences of temporal scene events has been constructed. Associated with the grammar is a set of visual primitives which are paired with the corresponding scene events. The detection of one or more visual events provides constraints on the sentences of scene events that can be generated by the grammar and consequently on the subsequent visual events that are likely to be observed. We demonstrate our approach in the domain of a table top breakfast scenario.

We shall show that our method does not preclude instances of unexpected events to occur. Rather, the scene event grammar, triggered by observations at any particular instance of time, imposes a partial ordering of the possible future events (appearance/disappearance of objects, object interactions). In this manner we can constrain visual expectations. The ordering could also be used in generating visual goals for vision system control purposes. This latter benefit will be reported elsewhere.

The paper is structured as follows. In the next section the vision system developed is briefly described. The table top breakfast scenario and its grammatical model are introduced in Section 3. The experiments performed to test the system are described in Section 4. Finally, in Section 5, conclusions are drawn.

2 Vision system architecture

The modules for scene evolution description and modelling add a high-level layer to the architecture of the VAP vision system. In this section we briefly overview the structure and operation of the system. Figure 1 depicts the basic building blocks. Attributed models of objects (shape, colour, texture, mobility) are stored in the (static) **object database**. The **scene description** database contains the internal model of the current state of the surrounding environment (the scene) and of the

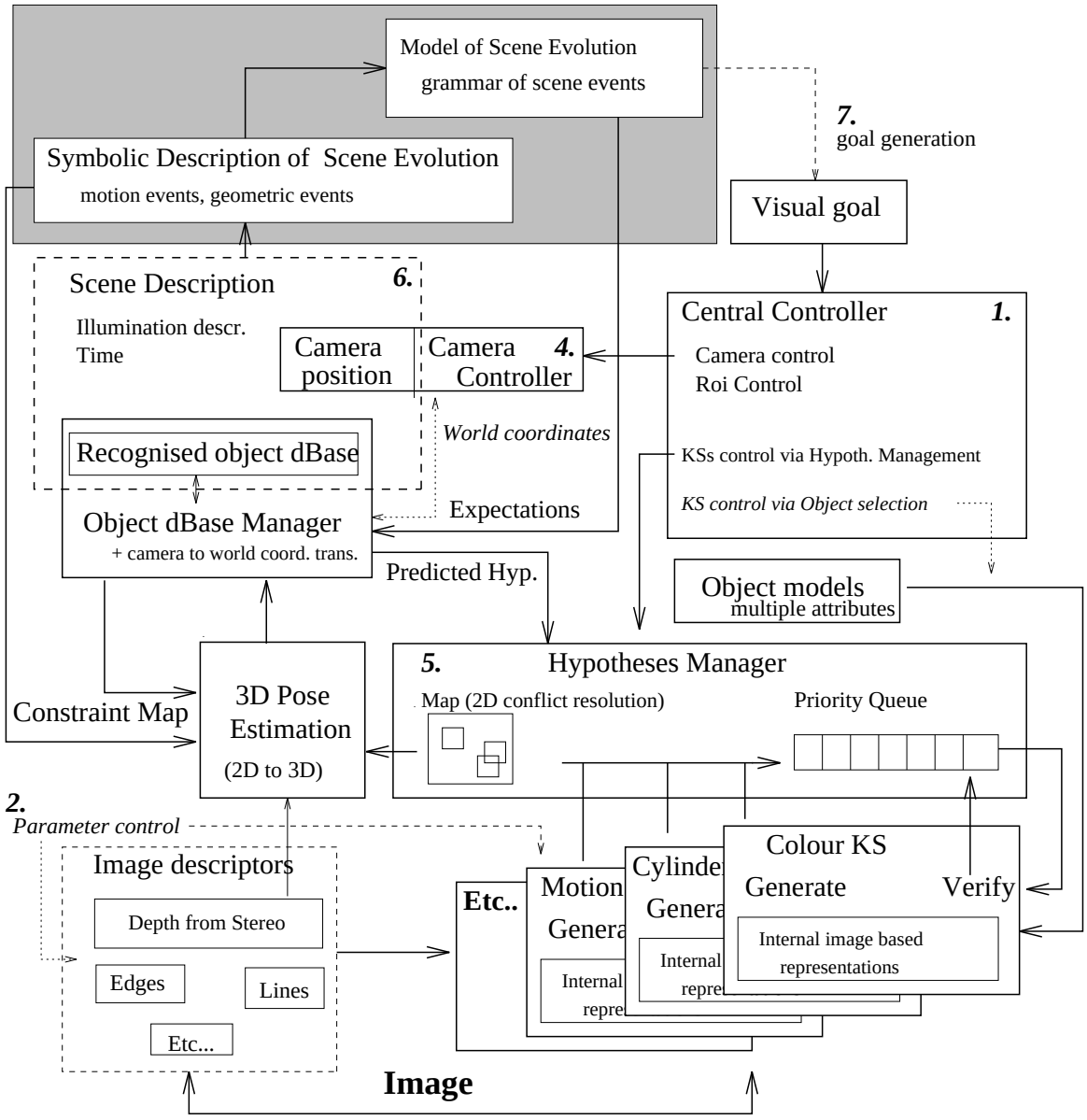

Figure 1: Structure of the VAP scene interpretation module

sensor, ie:

1. list of recognised objects in the scene with info about object 3D pose (in a camera-independent coordinate frame), confidence measure in the object hypothesis, time (of the first/last successful recognition etc.).
2. sensor description (eg. position, intrinsic parameters)
3. illumination description
4. real-time clock

The scene description is dynamic - new information from knowledge sources is constantly inserted in the database. Moreover, each entry in the scene description reflects our knowledge about the environment *at a given time*. Time stamping allows the system to asses the relevance of a piece of information to the current state of the external world. The **central controller** tunes the operation of the system to the externally defined visual goal by issuing commands to the **camera strategy unit**, **hypothesis manager** and individual **knowledge sources**. The camera strategy unit controls parameters of a stereo camera head (position, look point, zoom etc.) in order to acquire images in a way that helps to satisfy the current goal.

Knowledge Sources operate on the input image or on low-level image features implementing atomical recognition and segmentation strategies. Knowledges source communicate through hypotheses that correspond to regions of the image or groups of features belonging to the same object. Because we believe that the most efficient strategy for a any given object strongly depends on *context* of the task, there are no predefined strategies for recognition of objects. Instead, each knowledge source estimates the quality of its own hypotheses. The central **hypothesis manager** maintains a priority queue of hypotheses. High confidence hypotheses from the front of the queue are passed for verification to other knowledge sources. Efficient processing is achieved through this mechanism as computational resources are focused to regions likely to be knowledge sources. A sub-optimal strategy thus *emerges* as a result of a cooperative behaviour. The same priority queue is used for top-down *predictions*. Projections of objects previously detected can be placed in the priority queue and verified.

The **3D Pose Estimation** module combines the information from a number of sources (eg. stereo-based depth map, knowledge about 3D pose of previously detected objects) with the information contained in the hypothesis to establish its 3D pose. The result is inserted in the scene description database by the **Object Database Manager.**

3 Scene evolution: description and modelling

The premise behind our approach to scene interpretation is that the world we observe is structured both spatially and temporally. Whereas a spatial order is taken

for granted and has been the subject of modelling for computer vision purposes for decades, the role of temporal evolution in scene understanding is much more subtle. Yet if we consider any domain of activity that we may wish to observe, a surprising regularity in the evolutionary structure of scene events is revealed.

In order to illustrate how scene evolution may impact on its understanding let us consider one specific example: a table top breakfast scenario. The word breakfast immediately brings to mind the set of objects that we are likely to encounter when observing a breakfast scene: Cups, saucers, tea or coffee pot, milk jug, cutlery, sugar bowl and a box with cereals. The spatial relations of these objects is determined partly by their functionality and partly by conventional rules. But observing such a scene over a period of time one would also detect a distinct temporal structure. Part of a typical image sequence capturing and conveying evolutionary order is shown in Figure 2. Objects are constantly added, moved to interact or removed from the scene. There is a natural order to scene events. For instance, when tea is being served, the milk jug is likely to interact with a cup first, followed by the tea pot and finally the sugar bowl and spoon.

A breakfast scene evolution can be described by a sequence of scene states each of which belongs to one of three classes: **static (S)**, **dynamic (D)** or **not in the field of view (N)**. In a dynamic state, one or more objects will be in motion which can be either arbitrary or regular. We are not particularly interested in the actual path through the 3D space a moving object takes. However, a regular motion, e.g. that of a spoon in a cup during stirring, may convey important information about the identity of the object undergoing it.

A transition between two consecutive states is an **event**. We shall call such events **dynamic** as they mark a qualitative change in the scene. Dynamic events can be further classified into different categories depending on the nature of the transition. For instance, the change from a dynamic state into a static state will signify an object placement. However, from the point of view of detecting object interaction, it is important to recognise another class of events, called **geometric** events, which flag vertical alignment of two objects, their coincidence or contact. For instance the point of interaction of a cup with a milk jug will be defined by the instance when the milk jug is above the cup and therefore vertically aligned. Any other 3D geometric relationship between the two objects will not assume any special significance. The table below lists the various types of events.

```
event --> geometric --> alignment  (vertical)
                    --> coincidence / contact
      --> dynamic   any change of state is an event (see table below)
             for D-1, D0 (object dynamic at t-1 and t (now))
             change in motion type

%---- names of dynamic events :
%---- -1 refers to previous frame, 0 to current
1. Slow entry:   object in N-1, object in D0
```

```
2. Fast entry:   object in N-1, object in S0
3. Slow exit:    object in D-1, object in N0
4. Fast exit:    object in S-1, object in N0
5. Place:        object in D-1, object in S0
6. Pickup:       object in S-1, object in D0
%---- Visible object not generating an event:
7. Stays stat:   object in S-1, object in S0
8. Stays dyn:    object in D-1, object in D0
%--------------------
```

A sequence of events can be conveniently modelled by a grammar in which events are nonterminal symbols and rewrite rules generate other nonterminal symbols. Such a grammar has been inferred manually from a learning image sequence of breakfast scene evolution and is given in standard BNF form below (due to lack of space only the rules for the setting and drinking phases of the plan are presented):

```
Context/Scenario:  Setting for Tea Break.

        Objects:        Cups, Saucers, Plates, Spoons, Teapot,
                        Sugarbowl, Milkjug, Clutter (non recognizeable objs)
        Macro Actions:  Setting the table, Pouring Tea, Adding sugar,
                        Adding milk, Stirring the tea, Drinking,
                        Refilling (Pouring - milk - sugar), Cleaning the table
        Detectable actions:
                        Contact / no-contact w. hand & obj of type X
                        Coincidence/alignment in table plane betw X/Y
                        Movement of X (time), Stopped at Y (2D)
                        Entered FOV, Disappeared from FOV
                        Stirring X, Pickup of X, Placement of X
        Field of View:  Table without people (i.e. we will not see or at
                        least ignore any humans in the field of view)

General_Plan    := <SETTING>, <DRINKING>, <CLEANING>

Sequences for setting the table

SETTING         := <SET-CUPS>, <SET-AUX>, <SET-POT>
SET-CUPS        := <SET-CUP1> | <SET-CUP2> | <SET-CUP3>
SET-CUP1        := <SET-SAUCER>, <SET-CUPS>, <SET-CUP>
SET-CUP2        := <SET-SAUCER>, <SET-CUP>, <SET-CUPS>
SET-CUP3        := <SET-SAUCER>, <SET-CUP>
SET-SAUCER      := enter-fov(saucer), place(saucer)
SET-CUP         := enter-fov(cup), alignment-xy(cup,saucer), place(cup)
SET-AUX         := <SET-AUX1> | <SET-AUX2>
SET-AUX1        := <SET-SPOONS>, <SET-MILKJUG>, <SET-SUGARBOWL>
SET-AUX2        := <SET-SPOONS>, <SET-SUGARBOWL>, <SET-MILKJUG>
SET-SPOONS      := <SET-SPOON1> | nil
SET-SPOON1      := <SET-A-SPOON>, <SET-SPOONS>
SET-A-SPOON     := enter-fov(spoon), PLACE-SPOON
PLACE-SPOON     := alignment-xy(saucer,spoon), place(spoon)
```

```
SET-SUGARBOWL   := enter-fov(sugarbowl), place(sugarbowl)
SET-MILKJUG     := enter-fov(milkjug), place(milkjug)
SET-POT         := enter-fov(pot), place(pot)

Sequences for drinking tea incl pouring of tea and adding of sugar/milk

DRINKING        := <DRINK> | <REFILL>
DRINK           := <POURING>, <DRINK-TEA>
REFILL          := <DRINK>, <REFILL1>
REFILL1         := <DRINK> | <REFILL>
POURING         := <POUR-BLACK-TEA> | <POUR-REG-TEA>
POUR-BLACK-TEA  := pickup(pot), POUR-CUP
POUR-CUP        := <POUR-A-CUP>, <NEXT-POUR-ACT>
POUR-A-CUP      := alignment-xy(pot,cup), rot+(pot), <FILLING>, rot-(pot)
FILLING         := nil (an unobservable action!)
NEXT-POUR-ACT   := <POUR-CUP> | place(pot)
POUR-REG-TEA    := <POUR-BLACK-TEA>, <FILL-AUX>, <STIRRING>
FILL-AUX        := <SUGAR-FILL> | <MILK-FILL> | <BOTH-FILL>
SUGAR-FILL      := pickup(spoon), alignment-xy(spoon,sugarbowl),
                   alignment-xy(spoon,cup)
MILK-FILL       := pickup(milkjug), alignment-xy(milkjug,cup), rot+(milkjug)
                   rot-(milkjug), place(milkjug), pickup(spoon),
                   alignment-xy(spoon,cup)
BOTH-FILL       := <SUGAR-FIRST> | <MILK-FIRST>
SUGAR-FIRST     := <SUGAR-FILL1>, <MILK-FILL>
MILK-FIRST      := <MILK-FILL1>, <SUGAR-FILL>
SUGAR-FILL1     := pickup(spoon), alignment-xy(spoon,sugarbowl)
                   alignment-xy(spoon,cup), place(spoon)
MILK-FILL1      := pickup(milkjug), alignment-xy(milkjug,cup), rot+(milkjug)
                   rot-(milkjug), place(milkjug)
STIRRING        := stirring(cup), <PLACE-SPOON>
DRINK-TEA       := <GET-CUP>, <SET-CUP>, <NEXT-SIP>
GET-CUP         := pickup(cup), leave-fov(cup)
NEXT-SIP        := <DRINK-TEA> | nil

Required visual information to enable execution/identification of the above

pickup(X)               Hand reaches for objs and obtains contact
enter-fov(X)            An obj of type X has entered the field of view
leave-fov(X)            An obj of type X has left the field of view
place(X)                The object X is put down and the hand is removed from
                        the object.
alignment-xy(X,Y)       The objects X and Y are aligned in the plane defined by
                        the table (i.e. x and y are coordinates in a system
                        coincident with the table) Some kind of threshold is
                        needed for this.
rot+(x)                 In an obj centered coord system the elevation of the
                        obj decreased (turned towards the table)
rot-(x)                 In an obj centered coord system the elevation of the
                        obj increased (turned away from the table)
```

4 Experimental results

Figure 2 displays in gray level, 28 images from a colour sequence in which objects from the breakfast scenario are placed on a table. The task of the vision system is to interpret this scene i.e. recognise the objects and the actions performed with them. Objects are recognised by a process of isolating regions of interest (using either colour object models[21] or by chromatic differencing with a known background reference image[22]) and then applying specialist object recognition knowledge sources[7, 8]. Figure 3 shows a typical segmentation result achieved by chromatic differencing.

Following image processing, a log of visual states and events is automatically derived. The description of the first 5 frames of the sequence is given in figure 4. The state of recognised objects are categorised as static or dynamic and their 2D and 3D positions are maintained where possible. 3D information is derived from a ground plane constraint which assumes that objects which are static and are not being picked-up must reside on the previously calibrated table-top plane. The information from the log is used by the grammar which interprets the appearance of objects, their states and events as part of the temporal sequence that constitutes setting up a breakfast scene. For example, following the setting of the saucer and the appearance of the cup it is possible to infer that the cup will be placed on the saucer. The alignment and subsequent contact of these two objects confirms this prediction. Similarly, later in the sequence the alignment of milk-jug and cup can be interpreted as the act of pouring milk prior to filling the cup with tea. Thus the grammar rules provide a framework within which the visual events can be interpreted but also provide temporal context which permits predictions of future states and events to be made. This means that subsequent visual processing can be constrained both in terms of the areas of the scene and the likely measurements and decisions which have to be made. One of the outcomes of parsing of the visual events using the grammar is a natural language description of the sequence. Figure 5 gives the natural language output for the first 28 frames which are shown in Figure 2.

5 Discussion and conclusions

In this paper we have shown how a grammar can provide a powerful tool for incorporating temporal sequence information into image interpretation. Although the work shown is still in its early stages, it clearly demonstrates the potential of the method. The long term goal of the research is the efficient exploitation of high level spatio-temporal context to provide predictions and thereby simplify and control visual processing.

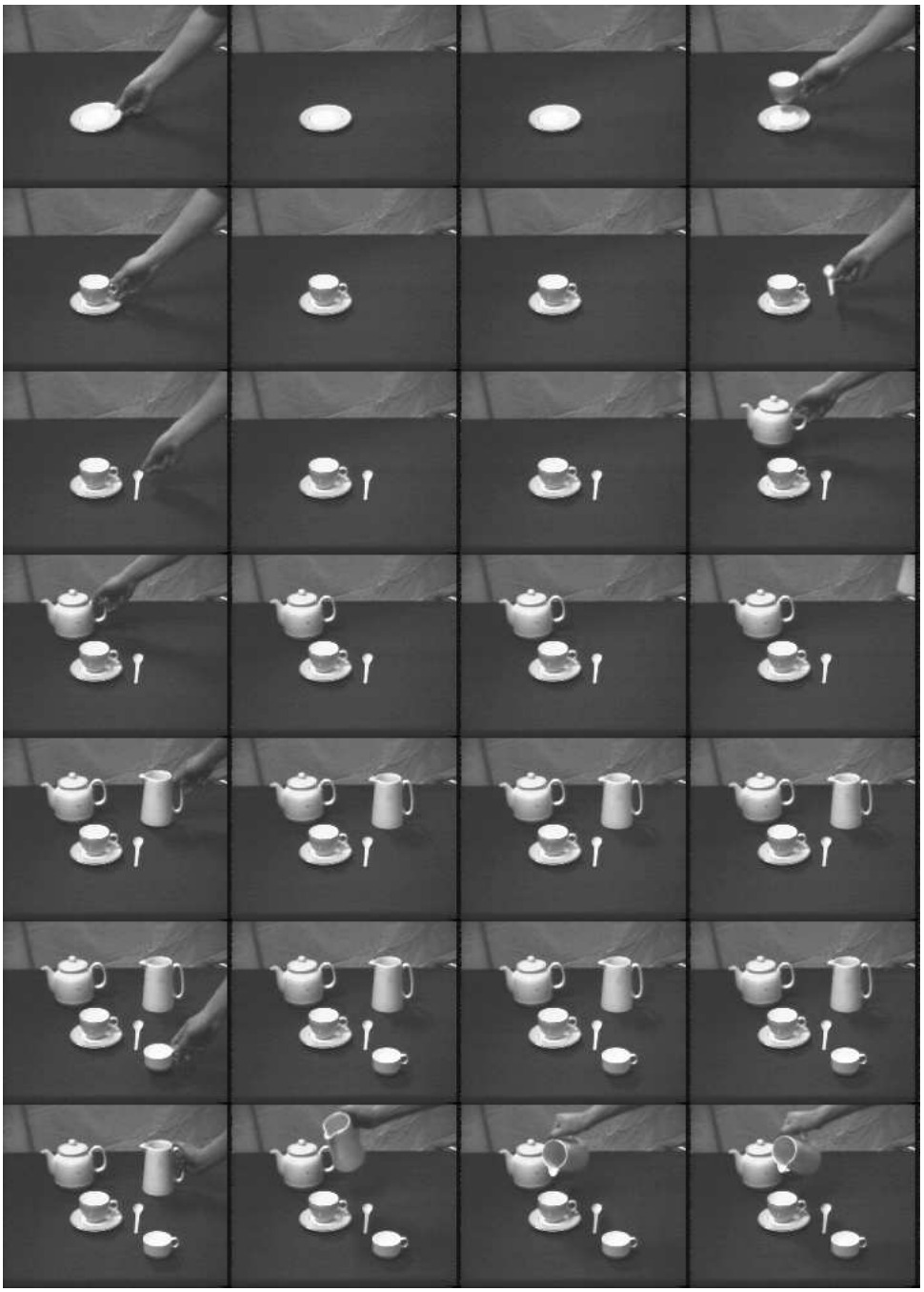

Figure 2: The first 28 frames of the Breakfast Scene Image Sequence

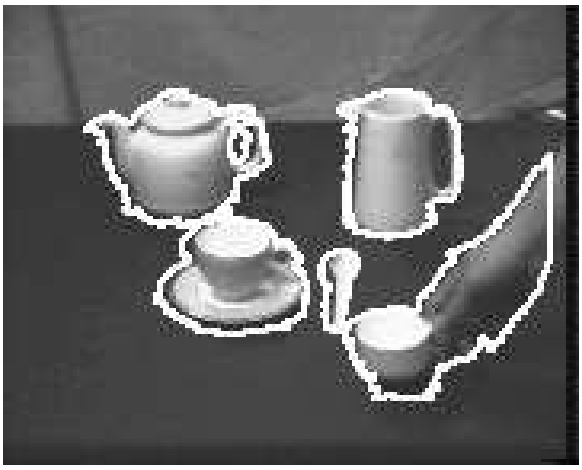

Figure 3: An example of the segmentation of objects via chromatic differencing

```
Frame 00:   Dynamic        saucer_1     72   88
Frame 01:   Static         saucer_1     71   87     456  374    0
Frame 02:   Static         saucer_1     71   87     456  374    0
Frame 03:   Static         saucer_1     71   87     456  374    0
            Dynamic        cup_1        72   74
            alignment(cup_1, saucer_1, 459, 274, 56)
Frame 04:   Static         saucer_1     71   87     456  374    0
            Dynamic        cup_1        71   90
            alignment(cup_1, saucer_1, 456, 374, 13)
    .         .              .
    .         .              .
```

Figure 4: Example of log of visual objects and events

```
Natural Language Description of Image Sequence:
-----------------------------------------------
NL: A saucer has been put on the table
NL: A cup was placed on the table
NL: A cup with saucer has been placed on the table
NL: A spoon was placed on the table
NL: the tea pot is here
NL: Milkjug is now on the table
NL: The sugarbowl is on the table
NL: The tea-break auxiliaries are now on the table
NL: The table has been set,
NL: The milk jug has been picked up
```

Figure 5: Natural Language description derived automatically from the grammar describing the breakfast scene.

Acknowledgements

This work was carried out as part of the European Union Basic Research Action project no 7108 "Vision As Process II".

References

[1] *First International Conference on Computer Vision, (London, England, 1987)* IEEE Computer Society Press.

[2] *Second International Conference on Computer Vision (Tampa, FL, 1988)* IEEE Computer Society Press.

[3] *Computer Vision - ECCV '90*. Springer-Verlag, 1990.

[4] Y Aloimonos, I Weiss, and A Bandyopadhyay. Active vision. In *First International Conference on Computer Vision, (London, England, 1987)*, pp 35–54.

[5] R Bajcsy. Active perception vs. passive perception. In *Proc. 3rd IEEE Workshop on Computer Vision*, 1985.

[6] A Blake and A Yuille. *Active Vision*. MIT Press, Cambridge, Massachusetts, 1993.

[7] M Bober, P Hoad, J Matas, P Remagnino, J Kittler, and J Illingworth. Control of perception in an active vision system. In *Intelligent Robotic Systems 93 (Zakopane)* pp 258-276, 1993.

[8] J Crowley and H I Christensen. *Vision as Process*. Springer-Verlag, 1994.

[9] J L Crowley, J M Bedrume, M Bekker, and M Schneider. Integration and control of reactive visual processes. In *Third European Conference on Computer Vision (Stockholm, Sweden)*, pp 47–58. Springer-Verlag, 1994.

[10] E. D. Dickmans. *Expectation-based Dymamic Scene Understanding*, chapter 18, pp 303–335. In [6], 1993.

[11] B. Draper, A.R. Hanson, and E.M. Riseman. ISR3: A token database for integration of visual modules. In *1993 DARPA Image Understanding Workshop*, pp 1155–1161, Morgan Kaufmann.

[12] B R Draper, A R Hanson, and E M Riseman. Learning knowledge directed visual strategies. In *1992 DARPA Image Understanding Workshop*, pp 933–940, Morgan Kaufmann.

[13] A. R. Hanson and E. M. Riseman. *Computer Vision Systems*. Academic Press Inc., Florida, 1978.

[14] LD T Lawton, T S Levitt, and P Gelband. Knowledge based vision for terrestrial robots. In *1989 DARPA Image Understanding Workshop*, pp 933–940, Morgan Kaufmann.

[15] D. G. Lowe. *Perceptual organization and visual reconstruction*. Kluwer, 1985.

[16] D. Marr. *Vision*. W.H.Freeman and Company, New York, 1982.

[17] J. Matas, P Remagnino, J Kittler, and J Illingworth. *Control of scene interpretation*. In [8], 1994.

[18] M. Mohnhaupt and B. Neumann. Understanding object motion: Recognition, learning and spatiotemporal reasoning. In *Towards Learning Robots*, pp 65–91. MIT, 1993.

[19] P L Palmer, H Dabis, and J Kittler. A performance measure for boundary detection algorithms. to appear in *Int Conf on Pattern Recognition (Jerusalem, 1994)*, IEEE Computer Society Press.

[20] P Remagnino, M Bober, and J Kittler. Learning about a scene using an active vision system. In *Machine Learning in Computer Vision*, pp 45–49, 1993.

[21] J. Matas, R. Marik, and J. Kittler. Illumination invariant colour recognition, submitted to *1994 British Machine Vision Conference*.

[22] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulea*. John Wiley, 1982.