Learning about a scene using an active vision system

P. Remagnino, M. Bober and J. Kittler
Department of Electronic and Electrical Engineering,
University of Surrey, Guildford GU2 5XH, United Kingdom

Abstract

An active vision system capable of understanding and learning about a dynamic scene is presented. The system is active since it makes a purposive use of a monocular sensor to satisfy a set of visual tasks. The message conveyed by the paper follows the thesis that learning is indispensable to the vision process to detect expected and unexpected situations especially when the monitoring of scene dynamics is employed. In such cases fast and efficient learning strategies are required to counterbalance the unsatifactory performance of standard vision techniques. The paper presents two distinct ways of learning. The system can learn about the geometry and dynamics of the scene in the usual active vision sense, by purposively constructing and continuously updating a model of the scene. This results in an incremental improvement of the performance of the vision process. The system can also learn about new objects, by constructing their models on the basis of recognised object features and use the models to predict unwanted situations. We suggest that the vision process benefits from the use of techniques for extracting scene characteristics and creating object models. As a consequence of the large variety of existing object classes a precompiled modelling of complex-shaped objects is unrealistic. Moreover, it is difficult to predict the presence and dynamics of all objects which may appear in the scene. Even if the creation of precompiled models for complex objects was feasible, the required recognition mechanisms would be slow and presumably inefficient.

1 Introduction

This paper presents the experience gained during the design and experimental evaluation of an active vision system [VAP89] [RJJJ], with the emphasis on its learning abilities. By learning we understand a multilevel process, the goals of which are multifold. At the highest level its aim is to construct a 3D symbolic model of the environment (learning about the scene). At the next level the system learns about the temporal context of scene objects. This information can be used to generate expectations about scene events that will result in increased efficiency of scene modelling. At the lowest level the system is able to learn about new objects. Object description is rendered in terms of geometric primitives which can be either 3D (generic polyhedra [KYJ93] and cylinders) or 2D (line segments, curves and their relationship), and attributes (colour and motion). We claim that a learning process is indispensable for the system i) to continuously improve its performance and ii) to cope with unexpected situations, for which pre-compiled knowledge is inefficient or non-existent. We present two experiments to show how the system can dynamically construct a symbolic scene model conveying information about the 3D space occupancy, and learn about new objects from the identification of observed image features.

The purpose of the paper is to introduce a system capable of learning the above three aspects by means of a coordinated action of knowledge sources. A system able to solve visual tasks in a partially known environment requires the cooperation of a number of knowledge sources, namely feature extraction, recognition and motion estimation modules. In order to solve a visual task the system performs a number of purposive actions which define its visual behaviour. Actions may involve static explorations of the environment or dynamic monitoring of events occurring in the field of view of the sensors. Usually actions are pre-compiled by encoding the knowledge of an expert about the analysed environment. However pre-compiled knowledge is not always sufficient to solve a visual task. There is a need for the system to learn about the environment and to respond effectively to dynamic events.

These mechanisms are incorporated in our system (see Figure 1). Learning about the scene is achieved by exploring the environment and building a symbolic model for it. This model is stored in the system database and can be re-used to simplify the execution of new visual tasks. Even more importantly, our system can cope with objects that are unknown. Objects are unknown because their appearance is not predictable at the design stage, or they cannot be easily modelled due to their complexity.

Constructing a model of the scene is a complex task. The system starts with an exploration of the scene where instances of known object classes are identified. The model of the scene is then built in terms of known objects and their 3D pose. The system may decide to invoke knowledge sources able to detect and segment motion in order to prevent object collisions. Motion is used for detecting and segmenting *unknown* 3D objects to which distinguishing image features are assigned. At a later stage, the introduction of unknown objects improves the understanding of the scene and simplifies its maintenance over time. The concept of unknown objects enables the system to *see* non-modelled object which would be invisible otherwise.

The next section describes the system structure and its modules. Implementation details are followed by an experimental section, and concluding remarks close the paper.

2 The Framework

The adopted framework uses a mixed control structure. At the highest level of the architecture decisions and relative actions are centralised. A central controller, the **Supervisor** [BHM+93] [PJJJ93], has the capability to take decision upon the optimal course of action needed to accomplish user defined visual tasks. Each visual task is decomposed into basic perceptual commands which determine the portion of visual space to be analysed, the sensor parameters and the sources of knowledge to be used. The supervisor adjusts the course of action each time a relevant event occurs in the field of view.

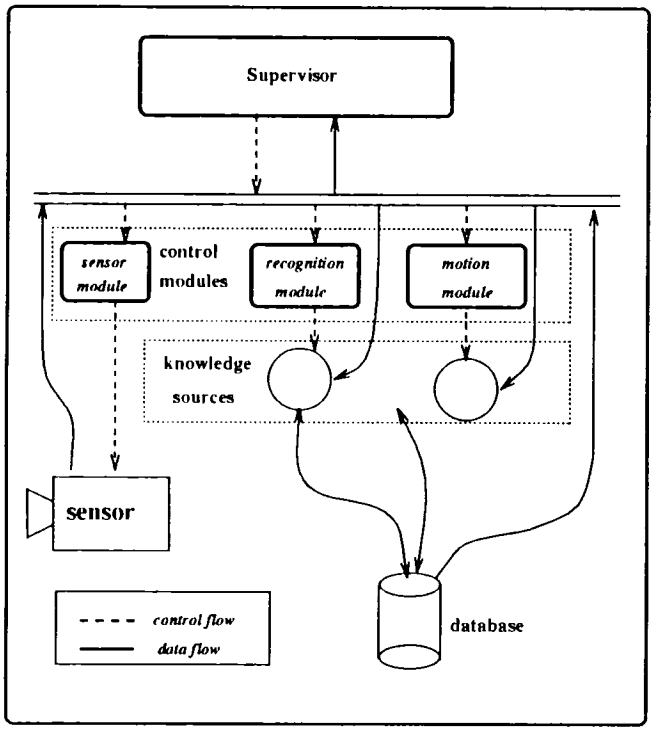

Figure 1: The architecture of the system

The lower levels of the architecture are controlled in a distributed fashion. Each perceptual command is executed and monitored by separate modules which are responsible for tight control of the sensor and the run-time parameters that are used by recognition and motion analysis processes. Knowledge sources that have been employed include object recognition modules, a model-based object tracker (which uses a Kalman filter predictor) and an HT (Hough transform) based motion extraction module [BK93].

Each module has a limited specialised knowledge of the world which enables the task execution without any subsequent intervention of the supervisor. For example the HT based tracking module can lock on and follow the object for many frames after a single request from the supervisor. Knowledge sources can communicate by accessing a common database as in the blackboard model [EM88]. Figure 1 shows the architecture of the system. Details concerning the employed knowledge sources can be found elsewhere [Rem93] [BK93] [KYJ93] [PJ93].

The proposed architecture follows designs of frameworks of other researchers in the field of Cognitive Science [AG89], Artificial Intelligence [HHW+89] and Vision [GB90]. The main characteristic of such frameworks is the distinction between *perception* and *cognition*. Perceptive and cognitive processes work closely first to interpret the sensed data as perceptual, and then to promote it to conceptual information. Similarly, our work proposes a distinction between the Supervisor module, responsible for the regulation of the vision course of action and its contextual situation assessment, and a number of independent modules which cooperate to interpret the viewed scene. The Supervisor is capable of reasoning; therefore it plays the role of the cognitive module. The knowledge sources responsible for the interpretation of the scene are assembled to form the perceptive module. One of the novelties of the architecture stems from the way visual tasks are encoded at knowledge source level. The supervisor defines a set of purposive actions which establish the viewing parameters; it also establishes which out of the recognition and motion analysis modules have to be activated. The existing knowledge sources work in a complementary fashion, since each of them has an independent responsibility (at the moment there are two recognition modules implemented, responsible for the identification of polyhedral and cylindrical objects). This simplifies the incremental construction of the scene performed by analysing object hypotheses created by the knowledge sources and combining new hypotheses with the already established ones.

Interpretation is identified with a *cycle of perception* which works at frame rate and it includes the scene imaging and its 3D geometric understanding. Reasoning is slower, intrinsically asynchronuous and driven either by events occurring in the field of view or by user defined tasks. It entails a continuous adjustment of the course of action depending on the detected event or the established interpretation.

Reasoning is represented as a heterarchical structure of tasks, where each user defined task is an independent *root* task, and the *terminal* tasks identify elementary visual operations. Reasoning evolves inside the supervisor in three major phases: i) a task handling phase, responsible for task decomposition, ii) a scheduling phase, dedicated to task activation, and iii) a response phase, in charge of the situation assessment.

Knowledge about the world is represented in terms of geometric relations and symbolic concepts which describe the scene in terms of objects classes and their attributes. Both symbolic and geometric representations are hierarchical. Geometric knowledge describes the scene in terms of a network of *reference frames*, identified with instances of reference objects which represent best the position of neighbouring objects. Symbolic knowledge describes the scene in terms of object classes. Classes are defined in terms of the functionality of objects, their attributes and their typical role in a scene.

Scene dynamics plays an important role in our system. The object mobility function, reflecting the likelihood that the particular instance of object is present in the scene, is modelled as an exponential function which decreases with time. A mobility factor is defined for each object to adjust the speed with which the exponential function decreases with time. The mobility factor is object and scene class dependent and the Supervisor may decide upon a different course of action according to the expected mobility of the objects known

to be present in the analysed scene.

The system is never idle. In case no user goal has been defined an initial exploration of the scene is performed by the Supervisor. Exploration is performed by directing the monocular sensor towards known reference frames, and exploring the surroundings. The supervisor has a prior knowledge of object classes which are typically found in the neighbourhood of reference objects. In this way a purposive course of action can be engaged in by looking for instances of such object classes. This generally involves the invocation of the most suitable knowledge sources and the selection of a restricted field of view to focus attention on particular portions of 3D space. Interpretation is typically guided to search for known objects in portions of space where the knowledge about the scene is scarce. The sensor is directed from one reference frame to another one. Searching, monitoring and all visual operations which involve geometric computations are performed locally to the active reference frame. This allows a considerable reduction of positioning errors, since each movement from one reference to another can be followed by a re-calibration of the sensing devise onto the local reference object. Besides, the use of a net of frames allows a more efficient exploration of the scene with a consequent improvement of the performance of the vision process.

Learning capabilities which have been built in have a different nature. On one side learning about the scene is accomplished in terms of known concepts. Currently the system has a prior knowledge of a number of specific object classes including their metric information; it can recognise generic cylinders and box-like structures; it understands motion and the concept of colour, and it can characterise objects by means of simple geometric primitives (straight lines, arcs and ellipses) and their groupings [BHM+93]. On the other side, learning is also accomplished by building new models for objects which are moving in the field of view.

3 Implementation details

The proposed architecture requires efficient processing algorithms at different levels of abstraction. At the lowest and intermediate level, it is necessary that the implementation routines process the sensed image and extract salient features. At the highest level, strategies are used to construct efficient reasoning mechanisms able to satisfy a set of visual goals and regulate their execution. The requirements are different at either level: image processing and feature extraction need a fast and flexible implementation, while reasoning calls for a more elaborate implementation able to cope with asynchronuous information flow and event-driven control flow. For this purpose image processing, feature extraction and motion analysis routines have been implemented in standard C, while the supervisor and the control knowledge has been implemented in CLIPS [GR89]. CLIPS, which stands for *C Language Integrated Production System*, is a rule based system implemented in the C language. Consequently, CLIPS is fast and it allows for a simple and efficient interfacing with low-level C routines. CLIPS inherits all the characteristics of a rule based shell. Rules can be grouped to form efficient and self-contatined software modules. Moreover, CLIPS version 5.1 has object-oriented facilities. Object classes, instances and inheritance can be defined and generated at run-time. The present solution may not be the most efficient, but it is a good starting point which proved to work in the presented system.

4 Experimental work

Both experiments that we are going to show have been carried out with real data. They emphasize the learning skills of our system.

The first experiment illustrates learning over time at a scene level. Figure 2 (*Learning about a scene*) presents a table top with two cups, a box and a plate. The system starts to perform a standard exploration of the table top and identifies only one of the two cups and the plate (the first figure corresponds to Frame 2). The second figure shows a later frame (Frame 4) where the visual behaviour has been changed to focus attention onto the identified cup. When focus of attention is activated the system performance is considerably improved since the employed knowledge sources only work on the region of interest set by the supervisor. The second figure shows that the plate instance is still supported since its mobility has been set to low values (this results in a long exponential decay). The third figure shows Frame 10. Focus of attention has been changed to the plate, which in the meantime has been slightly moved. Due to its low mobility, the cup instance is still supported in the database even if no longer visible in the field of view. Frame 16 is shown in the last figure. The cup instance has been removed from the database. The figure shows that the plate has been successfully tracked by the interpretation module.

The purpose of the second experiment is to illustrate how the system can learn about the environment and cope with unknown objects. The sequence in Figure 2 (*Learning about objects*) shows a table with a stationary Coke can and two moving mechanical toys (bugs). The initial goal of the supervisor is to explore the scene and learn about its contents. During this phase the Coke can is recognised by the knowledge source dedicated to perform the recognition of cylindrical objects. This is achieved by reasoning about the findings of low level routines such as edge detection, and line and ellipse finders [PJ93]. Once the Coke can has been recognised it is entered and maintained in the 3D scene model database. When motion is detected in the scene the supervisor switches its visual goal so as to check for possible collisions of moving objects with the can. To achieve it the supervisor activates the knowledge sources responsible for 3D motion modelling. Two moving regions are detected and their velocities and bounding boxes are determined (see Figure 2).

There is no model of bugs in the database, thus the moving objects cannot be recognised. At this stage the supervisor uses its capability to learn about the objects. This can be accomplished by collecting a set of features detected in the regions of interest. In the current implementation one of the features used to characterise a new object is its prominent colour. Other features, including shape and size are also used. We believe that the process of object description could be still improved if more features were employed. The moving objects (the bugs) are now labelled and corresponding features are placed in the database. This implies that they are now treated by the system as any other object that has been modelled. Our system has therefore learned to recognised two new objects, although at this stage it is unable to interpret them semantically.

The supervisor follows the moving objects and makes an attempt to determine how likely a collision with the Coke can is. The acquired 3D knowledge of the scene is used to map a 2D motion into 3D motion of the bugs on the table. The supervisor judges the danger of the collision on the basis of object 3D position and the direction of motion. In this particular experiment the danger is detected by comparing the angle between the collision route of a moving object and the actual motion direction against a predefined threshold. A white frame around the bug moving towards the can of Coke signals the danger of collision. The other bug is still monitored in case it changes its direction of motion. When the collision takes place the supervisor marks the colliding object (in our case the bug is identified by its colour) as potentially dangerous.

5 Conclusion and Future Work

We have presented an active vision system able to learn about the scene content and make use of its findings to cope with unexpected situations. The system is characterised by a distributed control structure where a set of knowledge sources are coordinated by a supervisor to solve given visual tasks. The system is never idle; if no event occurs the field of view is explored and the knowledge about the scene improved. If an object of interest is detected the supervisor may decide upon a course of action focused on the monitoring of eventual collisions. We have presented a technique upon which learning turns out to be indispensable to achieving a correct and reliable monitoring of the scene. We have shown the results of two of the many experiments conducted to demonstrate our technique and the importance of learning in a situation of surveillance. Although the framework has been proved valuable to solve visual tasks, its learning capabilities are still at a prototype level. Our intention is to augment its robustness and efficiency.

References

[AG89] V.V. Alexandrov and N.D. Gorsky. Expert systems simulating human visual perception. *International Journal of Pattern Recognition and Artificial Intelligence*, 3(1):19-28, 1989.

[BHM+93] M. Bober, P. Hoad, J. Matas, P. Remagnino, J. Kittler, and J. Illingworth. Control of perception in an active vision system: Sensing and interpretation. In *Workshop on Intelligent Robotic Systems 93*, pages 258-276, Zakopane, Poland, July 1993.

[BK93] M. Bober and J. Kittler. A hough transform based hierarchical algorithm for motion segmentation and estimation. In V. Cappellini, editor, *Proceedings on the 4th International Workshop on Time-Varying Image Processing and Moving Object Recognition*, Firenze, June 1993. Elsevier.

[EM88] Robert Englemore and Tony Morgan. *Blackboard Systems*. The Insight Series of Artificial Intelligence. Addison-Wesley, Stanford University, 1988.

[GB90] S. Gong and H. Buxton. Control of purposive-smart vision. Technical Report 561, University of London, July 1990.

[GR89] J. Giarratano and G. Riley. *Expert Systems, Principles and Programming*. PWS-KENT Series in Computer Science. PWS-KENT, Boston, 1989.

[HHW+89] B. Hayes-Roth, M. Hewett, R. Washington, R. Hewett, and A. Seiver. Distributing intelligence within an individual. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence*, chapter 15, pages 385-412. PITMAN, London, 1989.

[KYJ93] K.C.Wong, Y.Cheng, and J.Kittler. Recognition of polyhedral objects using triangle pair features. In *IEE Proceedings Part I: Communications, Speech and Vision*, volume 140, pages 72-85, February 1993.

[PJ93] P.Hoad and J.Illingworth. Recognition of 3D cylinders in 2D images by top-down model imposition. In *Proceedings of Scandinavian Conference on Image Analysis*, volume II, pages 1137-1144, Tromsø, Norway, May 25-28 1993.

[PJJJ93] P.Remagnino, J.Matas, J.Illingworth, and J.Kittler. A scene interpretation module for an active vision system. In *SPIE Conferenc on Intelligent Robots and Computer Vision XII: Active Vision and 3D methods*, September 1993.

[Rem93] Paolo Remangino. *Control issues in high level Vision*. PhD thesis, University of Surrey, University of Surrey, Guildford UK, July 1993.

[RJJJ] P. Remagnino, J.Illingworth, J.Kittler, and J.Matas. *Intentional Control of Camera Look Direction and View Point in an active Vision system*. Spring-Verlag. to appear in Vision as Process, published in ESPRIT Basic Research Action series.

[VAP89] Vision as process. Technical annex, ESPRIT-BRA 3038, European Commision, 1989.

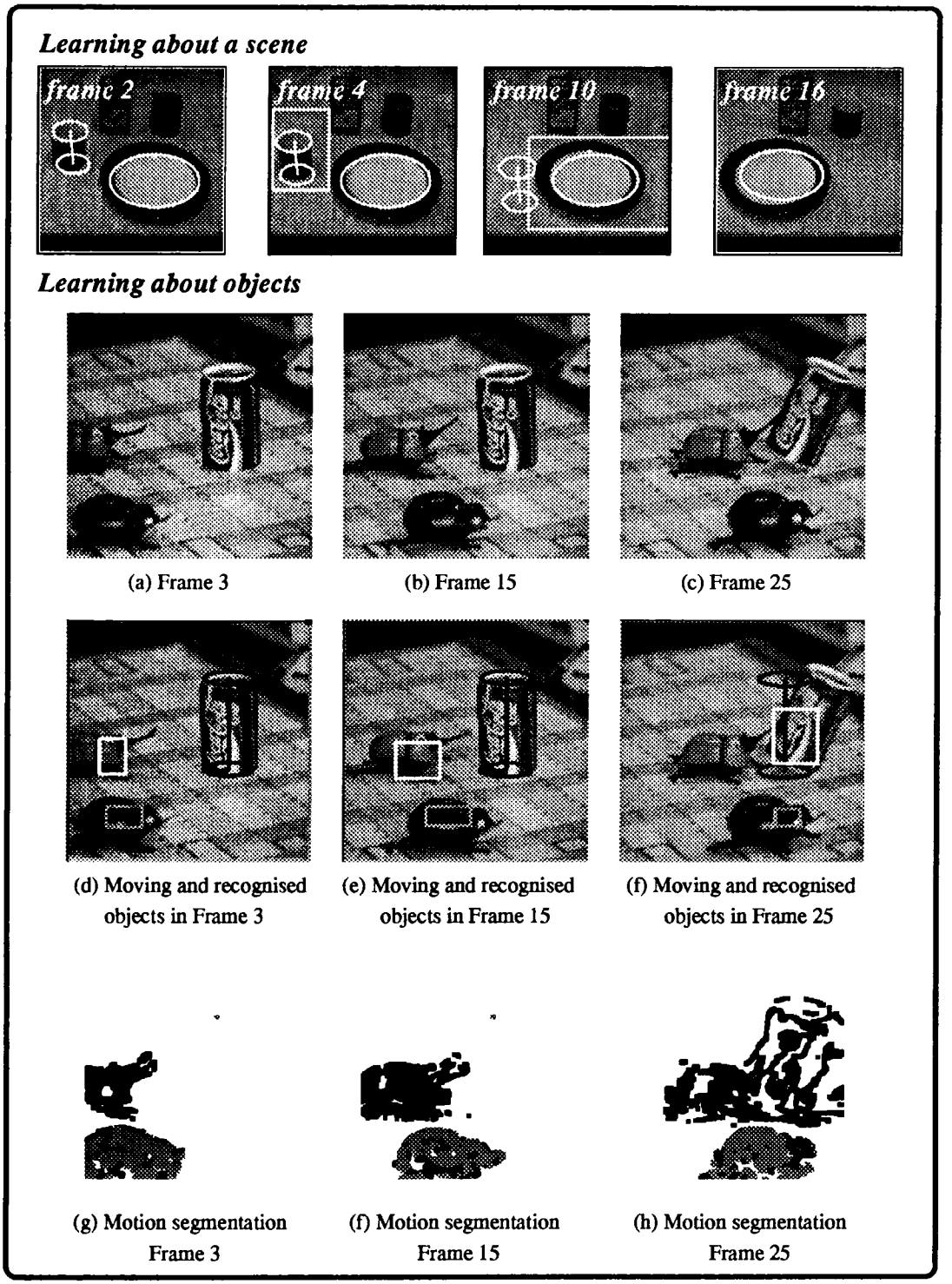

Figure 2: Learning about a scene and moving objects