M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 324
To appear in the IJCAI'95 Workshop on Entertainment and AI/Alife
April 1995

Intelligent Studios: Using Computer Vision to Control TV Cameras

Claudio S. Pinhanez and Aaron F. Bobick
pinhanez — bobick@media.mit.edu

Abstract

This paper demonstrates that automatic framing of TV shows is an interesting and tractable domain for both Computer Vision and Artificial Intelligence. Our basic goal is to build intelligent robotic cameras (*SmartCams*) able to frame subjects and objects in a TV studio upon verbal request from a TV director. To cope with the problem of relating visual imagery to symbolic knowledge about the scene, we propose the use of an architecture based on two levels of representation. High level *approximate world models* roughly describe the objects and the occurring actions. Low level *view representations* are obtained by vision routines selected according to the present state of the world, as described by the approximate models. The approximate world models are updated by contextual information extracted from the script of the TV show and by processing the imagery gathered by wide-angle, low-resolution cameras monitoring the studio. Our *Intelligent Studio* is composed of one or more SmartCams which share the representations of an approximate world model of the studio. A prototype has been implemented, and some examples of the cameras' responses to different requests are shown in the domain of a cooking show.

1 Introduction

The objective of this paper is to address the basic issues in the design and development of automatic TV cameras — *SmartCams*. By "automatic" we mean cameras operating without a cameraman which are able to generate TV-quality framing of scenes in response to simple requests from the director of the program. Presently it is already possible to find TV cameras controllable by joysticks that are used in studios where two or three cameras are controlled by one cameraman or even by the TV director himself. However, the jerkiness of the movements and the amount of work assigned to the single operator restricts their use to programs which can sacrifice quality for cost, such as on-line home shopping programs.

There are reasons to believe that the demand for TV programs is going to increase in the near future, particularly with the implementation of distribution centers of *video on demand* [8]. The prospect of having an on-line distributor of video in each small town, or each district in larger cities, opens new opportunities for TV programs dedicated to specialized and local communities.

In order to meet those demands, the costs and technical difficulties of the production of TV programs must be significantly reduced. One way is to decrease the need for highly trained professionals by using TV cameras and studios which require considerable less technical knowledge and are usable by people with only basic TV training. This simplification of the production process seems to require a technological shift similar to what happened in printing, where desktop publishing systems put the design process in the hands of the users, reducing printing costs and increasing design quality.

Editing systems are already converging towards simpler and less expensive computer-based systems. However, shooting costs are still high, and the use of TV cameras in effective ways is still a privilege of professionals. It must be understood that TV viewers are used to professional standards in TV shows and react negatively to bad framing and inappropriate use of television language.

A fundamental aspect of the problem is that a SmartCam must be able to relate the images from the studio set to the TV director's commands, employing knowledge about the structure of TV programs, human actions, and framing styles. We believe that research in this domain is interesting independent of the actual economic feasibility of SmartCams since the domain provides room for expanding the traditional approaches to machine vision by requiring the investigation of methods to correlate visual input with symbolic knowledge.

This paper begins by analyzing some of the basic difficulties in the construction of automatic TV cameras and arguing for the need of multi-level representations for the on-going actions. We then propose the use of *approximate world models* as a way to represent the subjects, objects, and actions happening in the studio. Approximate models make feasible the employment of real-time, simple computer vision algorithms, both to maintain the representation through time and to gather the information needed for framing. Then some specific architectural aspects are examined in more detail, and we conclude by showing some results obtained by a prototype SmartCam system framing a "cooking show".

2 Intelligent TV Studios

"Camera 3, ready for close-up of the chef. — More headroom. — Take camera 3. — Camera 1, ready for close-up of the bowl. — Take camera 1. — Follow the hands." This is how a TV director usually communicates with his cameramen in a TV studio. The TV director asks each camera for specific shots of the scene, or *calls*, and the "take" command signals that the camera is entering on

air. The instructions are brief and simple, tough they are clearly understood in the context of the scene.

After receiving a call, the cameraman looks for the appropriate subject, adjusts the framing, and waits, keeping the best possible framing. After the shot has been used, the cameraman receives a new call. This is the standard procedure for most TV programs including news, talk shows, sitcoms, and cooking shows. It is important to notice that such procedures are quite different from those used in movies or more elaborate TV programs where the images from each camera direction are shot separatedly and later assembled during editing.

Our *SmartCam* is conceived as a robotic camera for TV that can operate without a cameraman, changing its attitude, zoom, and position to provide specific images upon the verbal request of the TV director.

Framing is a quite complex task, requiring different kinds of information. For instance, a call for a close-up demands not only the information of the subject head's position and size but also the subject's direction of sight and the position of the eyes. Knowing the direction of sight is very important because profiles are always framed leaving some space in front of the face (called "nose room"). The height of the eyes is used, for instance, in a rule of thumb that states that eyes in a close-up should be leveled at two thirds of the height of the screen (see [17], pp.111–122, for this and other simple rules).

Moreover, framing changes according to the current action of the subjects. If an object is being manipulated, either it is fully included or it is put outside of the frame. Also, subjects in the background must be either framed completely or removed from the picture. The information required in these cases involves considerable understanding of the actions happening on the set.

However, each camera's information about the world is constrained by its current framing. A camera providing a close-up is unable to detect changes outside the image area, significantly reducing its ability to understand background activity. In order to cope with this problem, a simple solution is to use extra cameras in the studio whose only purpose is to monitor the scene. The resulting system, which we call an *Intelligent Studio*, is thus composed by coarse, fixed, wide-angle cameras watching the basic objects and actions and automatic, high-quality, mobile cameras blind to anything not inside their field of view, responsible for the generation of the show's images (see fig. 1).

It has been shown by Drucker [5] that framing can be mathematically modeled as a minimization process subjected to many constraints, provided that the system has precise knowledge about 3D shape, position and movement of all the objects. His work, however, is mainly concerned with highly mobile cameras, such as those modeled by computer graphics animations. Instead, we represent the framing requirements by rules, where the pre-conditions never refer to 3D data but to the projected image of the objects into the camera image plane. Using rule-based methods for framing, as described in the next section, has significant advantages in terms of simplicity and efficiency.

A "cooking show" is the first domain in which we are experimenting with our SmartCams. Framing cooking shows is quite difficult, since it involves many different kinds of close-ups of objects and actions. It is a considerably richer situation than those found in programs like news or talk-shows. Interestingly, it is not uncommon to see cameramen making mistakes during cooking shows because the chef's hands are quite unpredictable, there are lots of fast grabbing movements, and chefs often forget to turn to the scripted camera. In the following sections, examples drawn from the cooking show domain are used to illustrate some concepts, although most of the ideas apply to the general problem of framing.

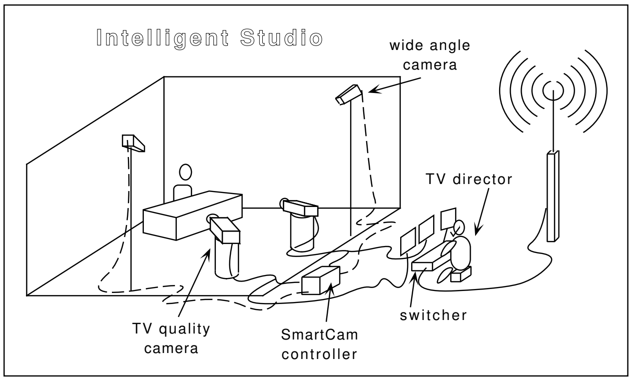

Figure 1: The concept of an intelligent studio, which uses wide-angle imagery to understand the events happening in the studio, and TV quality cameras to generate the images to be aired.

3 A Proposal: Approximate World Reconstruction

The task of designing SmartCams can be sub-divided into three basic problems: first, how to obtain information from the visual imagery; second, how to use that information for framing; and third, how to use a priori, symbolic knowledge in the process.

The traditional answer of computer vision research for the first two problems is what is known as the *reconstructionist approach* [9]. Basically, the imagery is extensively analyzed in order to three-dimensionally model the objects in the scene. Any task then refers exclusively to the reconstructed 3D-models to obtain information about the scene. The reconstructionist approach has been criticized on the grounds of feasibility [7], partially due to the fact that reconstruction requires precise algorithms which are quite susceptible to noisy images and to camera calibration. The appropriateness of 3D models and high level knowledge for task control is also subject of debate [3].

Purposive vision has been proposed as the alternative to the reconstructionist approach [1]. In this case, the vision algorithms are designed to provide answers for specific task requirements and the information gathered by such routines is hardly usable by any other task or module. The major problem with this approach is that it is difficult to incorporate high level knowledge into the system, compromising its robustness and adaptability. The reconstructionism vs. purposivism debate is still a central issue in computer vision (see [15] and the replies in the same issue).

Our proposal combines both approaches by approximately reconstructing the world, only to a level such that specific, purposive vision routines can be selected for each task. Contrary to traditional reconstructionism, our *approximate world model* is not a source of information for the tasks but, instead, used to select an appropriate set of vision routines, representational methods, and task controllers according to the current state of the world.

The motivation for our approach comes from the observation that many visual tasks can be performed using simple, view-based techniques if certain constraints are known to hold [14, 1]. For example, suppose one is trying to maintain the fixation of a camera on a moving person's head. If the system knows approximately where the head is within the view, then fast, simple motion- and template-based methods are sufficient to effectively track the head. Thus, a low-resolution, three-dimensional model of the person can adequately provide positional parameters for a vision routine; furthermore, the results of the routine can be checked against the projection of the 3D model into the image plane.

The computation of the approximate world model can also exploit common forms of contextual or semantic information beyond the limits of pure imagery analysis. Suppose, for example, that the system knows that at the current time a particular activity is taking place, e.g. "`the chef is wrapping the chicken`", and let us assume that the system is trying to maintain an approximate model for the location of the hands of the chef. The contextual knowledge of the action can be used to bias and/or verify the vision algorithms that determine the hand location. In this example, the hands would be searched for only in front of the body, above the waist, but below the shoulders.

Contextual information about actions can also be obtained from the *script* of the show. The script provides a reliable sequence of events, and delimits the basic vocabulary. A typical script of a cooking show is shown in fig. 2, in the same keyword form as it is used by the current version of our SmartCams. One important difference between the script of fig. 2 and a realistic one is that the start and finish times of each segment (keywords `:begin` and `:end`) are explicit times. Currently, the time information is manually obtained from the recorded sequence used to simulate SmartCams. To date we have not implemented action-trigger routines to detect the initiation and completion of an activity.

The present state of our system does not use a real robotic camera, but simulates one by using a moving window on a wide-angle image of the studio. In our experiment, a complete cooking show was recorded, using 3 cameras with wide-angle views, and digitized. Each camera sequence has 1200 frames of 680x480 pixels, 256 levels of gray, at a rate of 5 frames/second. A position of a camera is simulated by extracting a rectangular window of some size from the wide-angle images. Therefore, three parameters control the images output by a camera: the $< x, y >$ position of the center and the size of the window.

4 The Architecture

There are two fundamental kinds of representations used by the system, called *world* and *view* representations. View representations refer to how a particular camera sees an object. For instance, a camera can represent an object in the image plane as an irregular 2D blob or a planar ellipse, using only the image plane coordinate system.

```
(initialize-actions
 '((:begin 001 :end 061 :action-name "talk"
   :agent "chef" :patient nil :instrument nil
   :place nil :direction (:center))
  (:begin 062 :end 065 :action-name "turn"
   :agent "chef" :patient nil :instrument nil
   :place nil :direction nil)
  (:begin 066 :end 130 :action-name "talk"
   :agent "chef" :patient nil :instrument nil
   :place nil :direction (:side))
  (:begin 131 :end 134 :action-name "turn"
   :agent "chef" :patient nil :instrument nil
   :place nil :direction nil)
  (:begin 135 :end 275 :action-name "mix"
   :agent "chef" :patient "ingredients"
   :instrument "empty-bowl" :place nil
   :direction nil)
  (:begin 276 :end 336 :action-name "wrap"
   :agent "chef" :patient "chicken"
   :instrument "plastic-bag" :place nil
   :direction nil)
  (:begin 337 :end 380 :action-name "show"
   :agent "chef" :patient "chicken"
   :instrument nil :place nil :direction nil)
  (:begin 381 :end 400 :action-name "pound"
   :agent "chef" :patient "chicken"
   :instrument "meat-mallet" :place "chop-board"
   :direction nil)))
```

Figure 2: A typical cooking show script, written as the SmartCam system uses it. Keywords `:begin` and :end determine the start and finish times of each action.

World representations are related to the 3D world and approximately describe the occurring events and the shape and position of the objects. Currently, the system uses 3D cylinders and ellipsoids to represent objects, and symbolic representations (slots and keywords) for features like direction of sight and actions. The left column of fig. 3 shows some of the 3D world representations of the chef in the cooking show, projected in the wide-angle images, for three different frames of the sequence. The middle and right columns display view representations for the head and for the hands of the chef according to different cameras.

World representations play the role of approximate world models in the system. As we can see in fig. 3, the world information is not accurate enough to enable a proper framing. The sphere — the world representation of the head — is mis-aligned when projected into the image of the wide-angle camera. Framing routines use, in fact, the view representations shown in the middle and right columns of fig. 3.

The basic architecture for an Intelligent Studio composed of two SmartCams is illustrated in fig. 4, where the SmartCams can share the approximate world models since they refer to the exact same context — the actions in the studio. Mutually exclusive components of each SmartCam are surrounded by a dashed square. Basically, a camera module works in a feedback loop, fed both by requests from the TV director and from approximate world models' information. The task control module is responsible for controlling the "actuators" of the camera and selecting applicable and pertinent vision routines. The routines act upon the current images and the results are stored as view representations. The task control uses both the view representations and the world representations to select the next step of the system, always giving preference to the more accurate view-based data.

Maintaining the approximate model may require extra, non-task-related information, which in our system is extracted from the wide-angle images. However, the access to imagery, at the approximate level, was intentionally restricted to low resolution versions of the wide-angle images (170x120 pixels). In our simulated system the approximate world model manager uses images with the same direction of sight that the cameras. This is not a requirement of the proposed architecture, though quite a practical setup for a simulated system since it enables the use the same image source for both the wide-angle and the SmartCams.

The camera modules are, in principle, unaware of each other's existence and mutually exclusive knowledge. Also, in our off-line simulation of a real-time control system, the SmartCams do not have access to the full wide-angle image, but only to the window corresponding to the present camera field of view.

All the different representations of a given object are connected by its name, a lexical representation corresponding to a given visual entity in the world. These names are the primary index to the information known about a given object, and the same names are used by both the SmartCams and the approximate world model manager.

Knowledge about actions and events is also part of the world representation. In our current implementation a simple representation is used, quite similar to the script showed above. The knowledge required for framing is stored separately in procedural format. Framing objects require the satisfaction of many constraints, which demand different kinds of information from the view representations of the camera.

5 Using the World Model to Select Vision Routines

To better evaluate the idea of approximate models, we are currently using very simple, knowledge free (euphemism for "stupid") vision routines. All the vision methods are based on movement detection by frame subtraction, without calculating optical flow. The typical outputs of these methods are regions on the image plane marking the largest horizontally connected area where differences were found. Therefore, those routines can only be applied when there is enough confidence that the targeted object is in the field of view and it is moving.

For instance, consider the routine `region-uppermost-moving-blob` used to detect the head in close-ups, whose typical outputs are the irregular areas shown in fig. 3. This routine detects the largest moving blob in a pair of consecutive frames. Then divides the blob into upper and lower regions such that the horizontal length of each line of the blob is as uniform as possible with the average length of each region. The routine has no understanding about heads or people, but when applied within the right context, it can return an area corresponding to the head, which is the uppermost blob in the case of a motion-silhouette of head and shoulders.

The applicability conditions of `region-uppermost-moving-blob` when used as a head detector, are defined

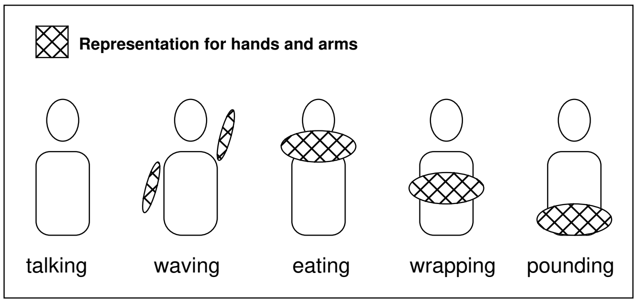

Figure 5: Different representations for the hands and arms are used according to the current action. When the subject is talking, no representation is necessary; when waving, each arm is represented separatedly, while in the case of manipulative actions both hands are represented by one big ellipsoid. Different manipulative actions also determine different expectations about the position of the hands relative to the trunk.

by the following rule:

```
IF projection of the approximate model of the target is
     partially contained in the camera current image AND
   size of the camera image is reasonable
     for a close-up AND
   there is no occlusion according to the
     approximate model
THEN apply region-uppermost-moving-blob
     to the last camera frames
```

In this manner, the same vision routine can be applied in different situations for a variety tasks, producing outputs with different meanings.

The approximate models are also used to check the output of the vision routines. In our case, many times the lack of object movement makes the movement-based routines return tiny, incorrectly positioned blobs. The task controller normally compares the area of the returned region with the projected area of the approximate 3D-model, and dismisses regions without plausible values.

The approximate representation of actions are used by many different routines at quite different levels. The manager of the world representations decides which is the best 3D model for a subject on basis of the action being performed by the subject. As shown in fig. 5, if a person is just talking, only a simple model composed by the trunk and the head is maintained. If the person is waving, each arm is represented by two different 3D models. Finally, if the subject is manipulating some object, then both hands are represented by an unique ellipsoid, corresponding to the area where the hand action is occurring. The expected initial position of the hands is also influenced by the current action, as demonstrated by the different heights for eating, wrapping and pounding in fig. 5.

Knowing the current action also changes the information required by a specific framing task. For instance, a medium close shot of a person who is manipulating some object must include the hands. However, the same shot of a talking or of a waving subject needs only information about the trunk and the head.

The idea of contextual information selecting appropriate vision routines has appeared before in other works [6,

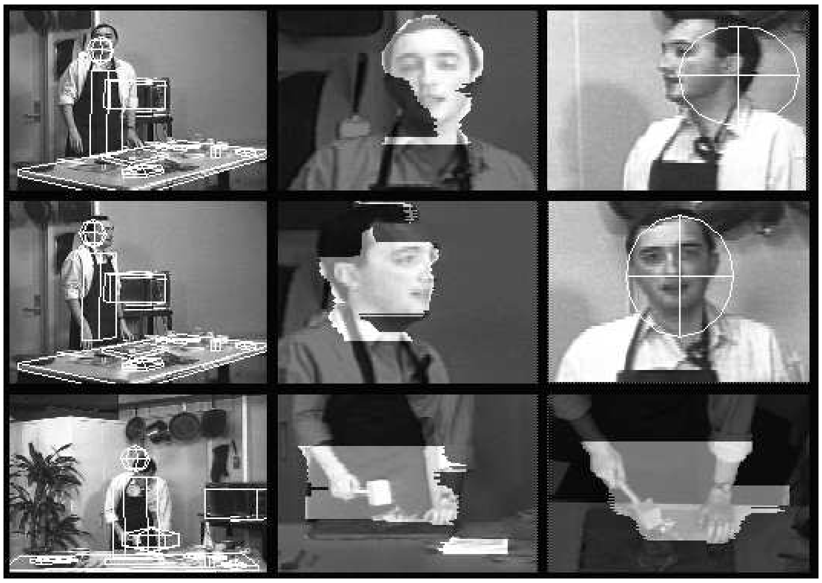

Figure 3: World and view representations: the left column contains the projection of the 3D-model representing the chef in some frames, and the middle and right columns present the view representations of the head or the hands according to different cameras. In the first and second rows, the cameras were asked to `‘‘close-up chef’’`; in the third row, the calls were `‘‘close-up hands’’`.

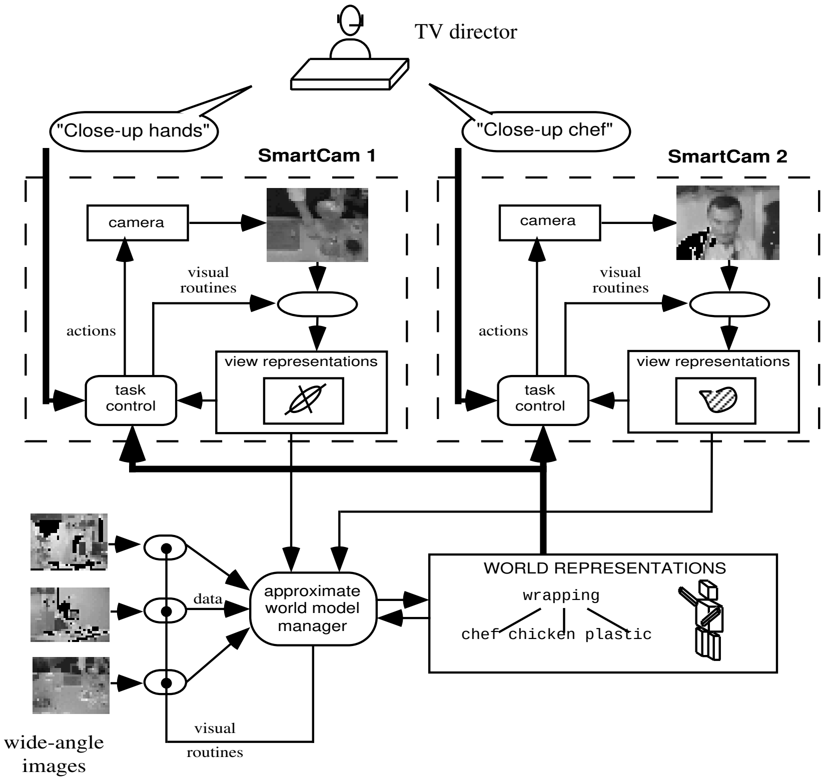

Figure 4: The architecture of a system composed by two SmartCams sharing the approximate world models of the studio. The dashed squares delimit the mutually exclusive components of each camera. The bottom part of the figure shows the structure of the modules responsible for maintaining the approximate world representations.

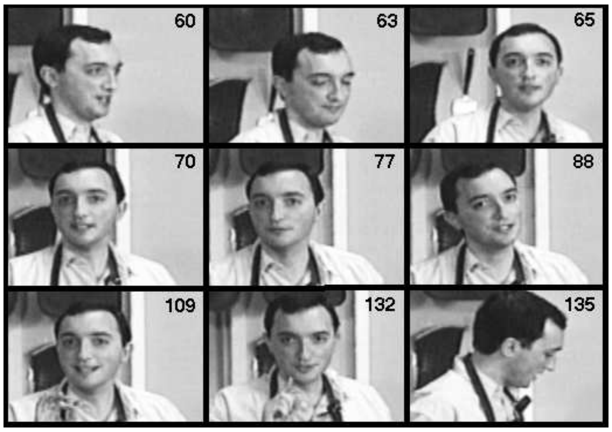

Figure 6: Response of the lateral camera to the call "`close-up chef`"; refer to the white bar in the background to observe the corrections performed by the camera.

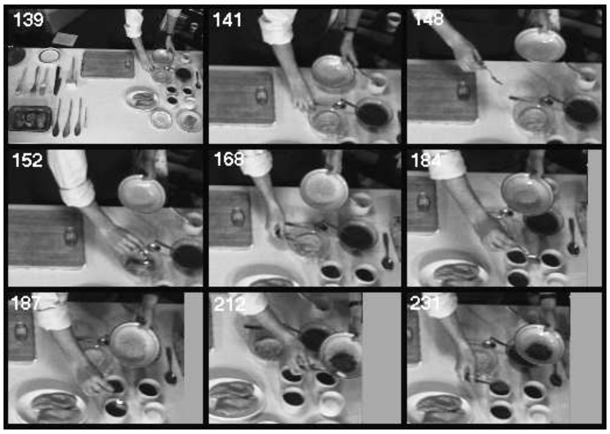

Figure 7: Response of the ceiling SmartCam to the call "`close-up hands`"; refer to the chopping board, on the upper left corner, to observe the corrections performed by the camera. The grey areas to the right of the last frames correspond to areas outside the field of view of the wide-angle image sequence used by the simulator.

13], which use simple, static descriptions of the world. As it is clear from the above examples, the combination of approximate 3D reconstruction and action representation can considerably expands the expressiveness of the pre-conditions for the applicability of a vision routine. For example, our system can verify if the *current* position and size of a group of objects satisfies a set of pre-conditions in the image plane. Therefore, the routine selection process is able to react opportunistically to the dynamics of the world.

6 Some Results

The current version of the system handles three types of framing (close-ups, medium close shots, medium shots) for a scenario consisting of the chef and about ten objects. The initial position of the human and the objects, and a timed script are given, and after that, any call can be made among the items in the vocabulary. Because of constraints on the speed of camera motion, and because adequate object motion sometimes needs to be observed for the vision routines to be effective, a camera typically needs 5 to 10 frames (1 to 2 seconds in the real video) to start finding view representations of the target object.

Figures 6, 7, and 8 show typical framing results obtained by the system. Figure 6 displays some frames generated in response to the call "`close-up chef`". The amount of correction made by the camera is better appreciated if we compare the position of the head with the white bar in the background. Frame 60 is an example of nose room: the camera provides space to the right because the chef is looking towards the right, according to the information in the script.

Figure 7 contains another sequence of frames, showing the images provided by a ceiling SmartCam tasked to provide "`close-up hands`". Initially framing the whole area (at frame 139), the camera zooms in into where the hands should be found according to the approximate 3D model. At frame 148 a view-based vision routine effectively detects movement in the area of the hands, producing the correction in the framing. The next frames of the sequence contain a great deal of small corrections, since the chef is taking ingredients from different bowls on the table; to see the corrections, the chopping board at the upper left corner can be used as a reference point. The grey areas to the right of the last frames correspond to areas outside the field of view of the wide-angle sequence used in the simulation.

Figure 8 is the response to a call for a "`medium-close-shot chef`". At frame 299 the camera is zooming out from the previous call, which had asked for a close-up of the chef. According to the script, the current action is manipulative (wrapping the chicken in a plastic bag), and thus a medium close shot must contain the hands of the subject, besides the head and the upper part of the trunk. At frame 312, when the chef is reaching for the chicken, the camera keeps this constraint as much as possible (refer to the microwave oven in the background to check the corrections). When the chef finishes wrapping and puts the chicken down, the camera zooms out to keep the hands inside the frame, as shown in frames 335 and 339.

A complete animated sequence of 400 frames (80 seconds), using standard calls and cuts of a cooking show, can be seen at the WWW-site:
`http://www-white.media.mit.edu/`
`vismod/demos/smartcams/smartcams.html`
The SmartCams in this sequence also use some simple rules which constrain the brittleness of camera movements, as it is the case in real TV for a camera "on-air".

7 Final Discussion

A first objective of this paper is to demonstrate that automatic framing is an interesting and tractable domain both for Computer Vision and Artificial Intelligence, and, especially, for the intersection between the two areas. We showed that the design of SmartCams seems to ask for solutions where vision routines can exploit a priori knowledge and general information about the state of the world, and that the ability to recognize and use action information is crucial.

Figure 8: Response of the lateral SmartCam to the call "`medium-close-shot chef`"; refer to the microwave oven in the background to observe the corrections performed by the camera.

Our basic idea is to construct an approximate world model used to determine which vision routines, representations, and control mechanisms should be applied in the current situation. We employ a multi-representational system (reminiscent of [2]) where the right representation for a given object is selected depending upon the task and the situation. Our proposal falls between the strict reconstructionist and purely purposive strategies currently debated in the community [1, 11]. We have demonstrated encouraging results in the domain of the Intelligent Studio, using SmartCams to interactively respond to a director's request for particular shots of a cooking show.

One of our major goals is to eliminate explicit time references from the script. The goal is have the script as a description of the sequence of events that are likely to occur, and a system able to identify the start and the end of actions, and to recognize them. The findings of Newtson *et al.* [10] and Thibadeau [16] show that subjects largely agree about the segmentation points between different actions, and that the expectation about the next action plays a fundamental role in the recognition and in the segmentation processes. Thus, a non-timed version of the script would theoretically give most of the information needed, though we believe that implementing action segmentation and recognition is still a considerable task.

Another clear direction of improvement is to design a language which describes the pre-conditions and the outputs of vision routines in a domain-independent way, enabling the easy incorporation of new routines to the system and the re-use of vision routines in the case of completely new domains. The work of Prokopowicz *et al.* [12] examines some typical characteristics of such a language. The use of approximate world models can significantly increase the expressiveness of such languages.

For commercial SmartCams, a major concern is certainly real-time processing. The project ALIVE [4] has shown that real time scene understanding is possible in very constrained domains, with basic vision routines of complexity similar to ours. Our present system runs about one order of magnitude slower than real-time, without using any special hardware for vision processing and doing the reasoning processing in LISP. Considering that maintaining the approximate world models has moderate processing power demands, we believe it is possible to implement real-time SmartCams by using image-processing dedicated hardware, as we intend to do.

References

[1] John Yiannis Aloimonos. Purposive and qualitative active vision. In *Proc. of Image Understanding Workshop*, pages 816–828, Pittsburgh, Pennsylvania, September 1990.

[2] Aaron F. Bobick and Robert C. Bolles. Representation space: An approach to the integration of visual information. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 492–499, San Diego, California, June 1989.

[3] Rodney Brooks. Intelligence without reason. Memo 1293, MIT A.I. Lab., April 1991.

[4] Trevor Darrel, Pattie Maes, Bruce Blumberg, and Alex Pentland. A novel environment for situated vision and behavior. In *Proc. of CVPR-94 Workshop for Visual Behaviors*, pages 68–72, Seattle, Washington, June 1994.

[5] Steven M. Drucker. *Intelligent Camera Control for Graphical Environments*. PhD thesis, MIT, Program in Media Arts & Sciences, 1994.

[6] Daniel D. Fu, Kristian J. Hammond, and Michael J. Swain. Vision and navigation in man-made environments: Looking for syrup in all right places. In *Proc. of CVPR-94 Workshop for Visual Behaviors*, pages 20–26, Seattle, Washington, June 1994.

[7] Ramesh C. Jain and Thomas O. Binford. Ignorance, myopia, and naiveté in computer vision systems. *CVGIP: Image Understanding*, 53(1):112–117, January 1991.

[8] David Kline. Align and conquer. *Wired*, pages 110–115;164, February 1995.

[9] David Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *Proc. R. Soc. Lond. B*, volume 200, pages 269–294, 1978.

[10] Darren Newtson, Gretchen Engquist, and Joyce Bois. The objective basis of behavior units. *Journal of Personality and Social Psychology*, 35(12):847–862, December 1977.

[11] Claudio S. Pinhanez. Controlling a highly reactive camera using a subsumption architecture. In *Proc. of Applications of AI 93: Machine Vision and Robotics*, pages 100–111, Orlando, Florida, 1993.

[12] Peter N. Prokopowicz, Michael J. Swain, and Roger E. Kahn. Task and environment-sensitive tracking. In *Proc. of CVPR-94 Workshop for Visual Behaviors*, pages 73–78, Seattle, Washington, June 1994.

[13] Raymond Rimey and Christopher Brown. Control of selective perception using bayes nets and decision theory. *IJCV*, 12(2/3):173–207, 1994.

[14] Thomas M. Strat and Martin A. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *IEEE PAMI*, 13(10):1050–1065, October 1991.

[15] Michael J. Tarr and Michael J. Black. A computational and evolutionary perspective of the role of representation in vision. *CVGIP: Image Understanding*, 60(1):65–73, July 1994.

[16] Robert Thibadeau. Artificial perception of actions. *Cognitive Science*, 10:117–149, 1986.

[17] H. Zettl. *Television Production Handbook*. Wadsworth Publishing, Belmont, California, 4th edition, 1984.