M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 329
To appear in the IEEE Workshop on Context-Based Vision (ICCV'95)
This paper is a revised version of TR #313 (January 1995)
April 1995

Using Approximate Models as Source of Contextual Information for Vision Processing

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

Abstract

Most computer vision algorithms are based on strong assumptions about the objects and the actions depicted in the image. To safely apply those algorithms in real world image sequences, it is necessary to verify that their assumptions are satisfied in the *context* of the visual process. We propose the use of *approximate world models* – coarse descriptions of objects and actions in the world — as the appropriate representation for contextual information. The approximate world models are employed to verify the applicability of a vision routine in a given situation. Under these conditions, a task module can reliably use the outputs of the contextually-safe vision routines, without having to refer to an accurate reconstruction of the world.

We are using approximate world models in a project to control cameras in a TV studio. In our *Intelligent Studio* automatic cameras respond to verbal requests for shots from the TV director. Contextual information is obtained from the script of the TV show and from the imagery provided by wide-angle, low-resolution cameras monitoring the studio. Some examples of the cameras' responses to different requests are shown in the domain of a cooking show.

1 Introduction

It is our belief that computer vision systems should employ a large repertoire of vision routines with different constraints and assumptions, as a way to cope with the richness and unpredictability of the world. In this work, we propose some representations for contextual information which enable a vision system to dynamically select and tune appropriate vision routines according to the (believed) state of the world.

We define the *context of a vision system or routine* as all the information about the world used by the system or routine, except the imagery. Geometric models, previous results, and a-priori statistics are examples of contextual information used in vision processing.

Our general objective is not only to represent context in order to improve the internal performance of particular vision routines, but, more importantly, to ensure that the assumptions behind the particular vision routines are satisfied. By representing appropriately the information about objects, human beings, actions, and events of the world, it is possible to estimate whether a vision routine is likely to fail in a given image. For instance, if the context representations contain information about depth and shape, it is possible to check for occluded objects and to avoid the calling of routines sensitive to occlusion.

Our hope is to be able to cope with one of the recurring criticisms (e.g. [9]) of much of computer vision, that many of the developed techniques are brittle, functioning well only if some set of restrictive assumptions about the situation are true. These assumptions may be considered "restrictive" if they are often false. One example is the Lambertian shading model assumed by many shape-from-shading algorithms ([15, 7]). In fact, it is difficult to construct a surface whose reflectance is nearly Lambertian, making the assumption all that more suspect. However, knowledge about the surfaces and about the approximate position and attitude of an object can assure that Lambertian-based routines have a chance to succeed in the area of the image plane corresponding to the object.

The basic idea is that the vision system maintains an **explicit, global, approximate** model of the scene. This model may include geometrical models, categorical spatial information, as well as general action and event specifications. Moreover, we believe it is necessary to use multiple representations for the same object of the world, both in order to be able to provide information according to specific needs, and also as a way to handle contradiction. Although our work is similar to Strat and Fischler's ([20]) in the use of contextual information, it differs significantly in the way context is represented.

In the remaining sections we discuss more fully the type of approximate models we are proposing and describe how those models are used as context. We then describe an example domain and task — the Intelligent Studio — where an approximate model representation of context is employed. We present the architecture developed to simultaneously maintain an approximate global model while performing perceptual tasks using view-based techniques. Finally, the competence of such a system is demonstrated within the Intelligent Studio domain.

2 Approximate Information as Context

The goal of using context to increase the robustness of vision operations is becoming a more common one ([6, 5]). The difficulty is always how to represent contextual information and to allow that information to bias vision computations. Our proposal for incorporating context into a vision system is to allow context to impact an *approximate world model* which is used in turn to select and validate vision computations.

Our emphasis in the approximate character of the world model is related to the own nature of context. First, it reflects the limitations in the access to the "true" reality of the world. But mostly, approximate is understood as "insufficient" and "imprecise", as information able to guide the selection of vision routines, but not to be used directly in the accomplishment of a task.

Part of the motivation for our approach comes from the observation that many vision tasks can be performed using simple, view-based techniques if certain constraints are known to hold ([20, 2]). 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 approaches are sufficient to effectively track the head. Thus, a low resolution, inaccurate, 3D model of the person could adequately initialize such a routine, and furthermore the results of the view-based routine could be checked against the 3D model.

Similarly, context can be provided as categorical spatial information. For example, the information that a person is looking "forward" with respect to a camera interested in recognize a person would enable the system to select a face-based recognition algorithm based upon principal components of image appearance ([13, 3]).

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

We define the *applicability conditions* of a vision routine to be the set of assumptions, that, if true in the current situation, warrants faith in the correctness of the results. If the applicability conditions of a vision routine X are implied by the context, then we say that *the applicability conditions of* X *are satisfied*, and the outputs of X can be accepted. Moreover, the output of view-based routines can be checked against the approximate models, with approximate agreement being required to accept the results. In this case, the applicability conditions of the vision routine contain post-conditions: constraints on the required agreement between the computed results and the approximate model.

In our proposal, the primary function of the approximate world model is to check the applicability conditions of view-based routines. The approximate world model is not to be used directly as a source of the perceptual information required by specific tasks.

It is interesting to situate our scheme in the ongoing debate about reconstructionist vs. purposive vision (see [21] and the replies in the same issue). We are arguing that reconstruction should exist at the approximate level, guiding the purposive vision routines of the view-based level of representation. By making the task routines dependent mainly on view-based data, we avoid the theoretical and pragmatical trap of reconstructing the world accurately. And by building approximate models of the objects, we can avoid the danger of depending solely on task specific vision routines which do not work reliably in all situations.

Two final points: first, we note that, like Strat and Fischler ([20]), we are taking a "compiled" approach to contextual reasoning. By compiled we mean that we are eliminating the need for highly developed qualitative reasoning system sometimes envisioned for the processing of highly abstract contextual information ([11]).

Consider, for instance, a simple contextual description like "`Mary is wrapping a gift`". In the approach we are proposing, the system has rules which directly relate the action of wrapping to the possible location and motion of the hands with respect to the body. The effect of this rule is that if the approximate location of the body is known, then the approximate location of the hands is known as well. This knowledge in turn is used to select and initialize hand tracking procedures.

Second, maintaining an approximate world model requires additional sensing and computation which might not be required to directly address current perceptual tasks. We are willing to incur this additional cost to increase the competence of the task-based routines.

3 A Testbed: Framing for TV

Our approach of using approximate world models is being developed in a system we are constructing for the control of TV cameras. The ultimate objective is to develop a camera for TV that can operate without the cameraman, changing its attitude, zoom, and position to provide specific images upon request. We call these robot-like cameras *SmartCams*.

In a normal TV studio, there are two or more cameras, connected to the switcher, the device used by the TV director to select the image to be put "on air". The main *framing* decisions — the position, attitude, and zoom of each camera — are also made by the TV director, who asks each cameraman for specific shots of the scene, or *calls*, determining the subject or object to be framed and the approximate size. Typical examples of the communication between the TV director and the cameramen are sentences like "`close-up newscaster`", "`medium-shot`", "`close-up product`", "`zoom more tightly`", "`more head-room`".

After receiving a call, the cameraman looks for the

```
(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 1: 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.

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.

Framing is much more difficult than tracking and requires much more 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. Direction of sight is important because profiles are always framed leaving some space in front of the face (called "nose room"). The height of the eyes is used in a rule of thumb stating that eyes should be leveled at two thirds of the height of the screen (see [25], pp.111–122, for this and other simple rules). Framing also requires knowledge about the current actions, as is exemplified by an example detailed later.

A "cooking show" is the first domain in which we are experimenting with our SmartCams. Framing cooking shows is quite complex, involving many different kinds of close-ups of objects and actions. Cooking shows are also interesting because the information concerning actions and objects is quite explicit in the form of the recipe turned into a TV script. The recipe provides a reliable sequence of events and delimits the basic vocabulary. A typical script of a cooking show is partially shown in fig. 1 in the same keyword form as it is used by the current version of our SmartCams.

One important difference between the script of fig. 1 and a realistic one is that the start and finish times of each segment (keywords `:begin` and `:end`) are explicit times. Currently, the frame numbers are obtained manually from the recorded sequence used to simulate SmartCams. To date we have not implemented routines to detect the initiation and completion of an action.

The present state of our system does not use real moving cameras, but simulates them using a moving window on a wide-angle image of the set. A complete cooking show was recorded using 3 cameras with wide-angle views and digitized. Each camera sequence has 1200 frames of 680x480 pixels, black & white, 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 Representing Context

It is important to distinguish representations of the objects in the world which are dependent of the particular point of view of a camera. We call those *view representations*, referring to how a particular camera sees an object according to the image plane coordinate system. For instance, a camera can represent a 3D object in the image plane as an irregular 2-dimensional blob or a planar ellipse. All vision routines return view representations and all the task-specific information is obtained from them.

View-independent representations are called *world representations.* World representations describe coarsely the shape and position of the objects, and the events happening in the world. Currently, the system uses 3D cylinders and ellipsoids to represent objects, and textual representations (slots and keywords) for features like direction of sight and actions.

Figure 2 shows the 3D world representations of the chef (in the cooking show) projected in the wide-angle images and the view representations for the head and for the hands according to two different cameras for three different frames of the sequence. Since the world model is assumed to be only approximate, there is no need for precise calibration of the cameras.

World representations play the role of approximate representations in the system and are the chief source of information for the routine selection process. As we can see in Figure 2, the world information is not accurate enough to enable a proper framing. The world representation of the head — a sphere — is mis-aligned when projected into the image of camera 1. Therefore, framing routines use the more accurate view representations such as those shown in the middle and right columns of figure 2.

All the different representations of a given object or event are connected by its name, a lexical representation corresponding to objects and actions. These names are the primary index to the information known about a given object. They also link the internal system information to the outside world, in our case, to the words used by the TV director to communicate with the SmartCams.

The knowledge required for framing is stored separately in procedural format. Framing objects in different situations require different rules which also de-

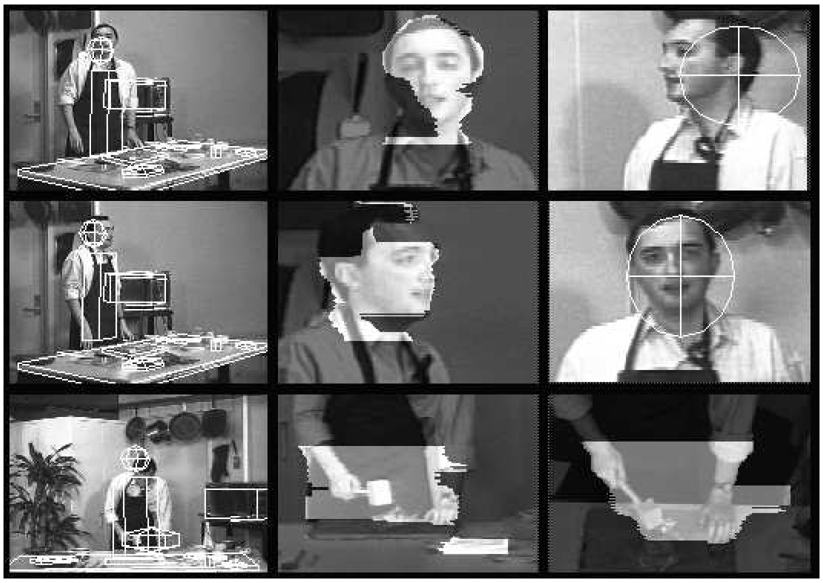

Figure 2: World and view representations: the left column contains the projection of the 3D-model representing the chef in some frames, and the middle and right columns present the view-representations of the head or the hands according to different cameras.In the first and second rows, the cameras were asked to "`close-up chef`"; in the third row, the calls were ''`close-up hands`".

mand different kinds of information from the view representations of the camera. Since different representations for an object can be needed in different situations and since each framing rule asks for specific information, we structured the system such that view representations are computed on demand. For example, the view representation of the hands are only computed if needed for a task. Similarly, approximate world models are only maintained when it is plausible that they are necessary. Again, hands might be tracked and modeled during an object manipulation task, but perhaps not when a person is simply speaking to the camera: at such times the hand models are normally not needed neither are various motion-based algorithms likely to succeed.

5 Architectural Issues

The basic architecture for a system composed of two SmartCams is illustrated in Figure 3. The 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 model's information. The task control module is responsible for controlling the actuators of the camera and for selecting applicable and pertinent vision routines. The routines act upon the current image and the results are stored as view representations.

Initially, the system is given the keyword form of the script and the initial shape and position of all objects (only at the approximate world level). The script itself does not provide adequate information to maintain reasonable approximate world models, what requires 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 purposivefully 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 as the SmartCams. 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 knowledge. Also, in our 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 condition. This follows our conception of an *Intelligent Studio* (fig. 4) which uses coarse, fixed, wide-angle cameras to monitor the basic objects and actions, and automatic mobile cameras blind to anything not inside their field of view, though able to ask the intelligent studio for contextual information.

6 Methods Sensitive to Context

To evaluate the idea of approximate models, we are currently using very simple (euphemism for "stupid") vision routines. All the vision methods are based on movement detection by frame subtraction, without calculating optical flow. The typical output 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.

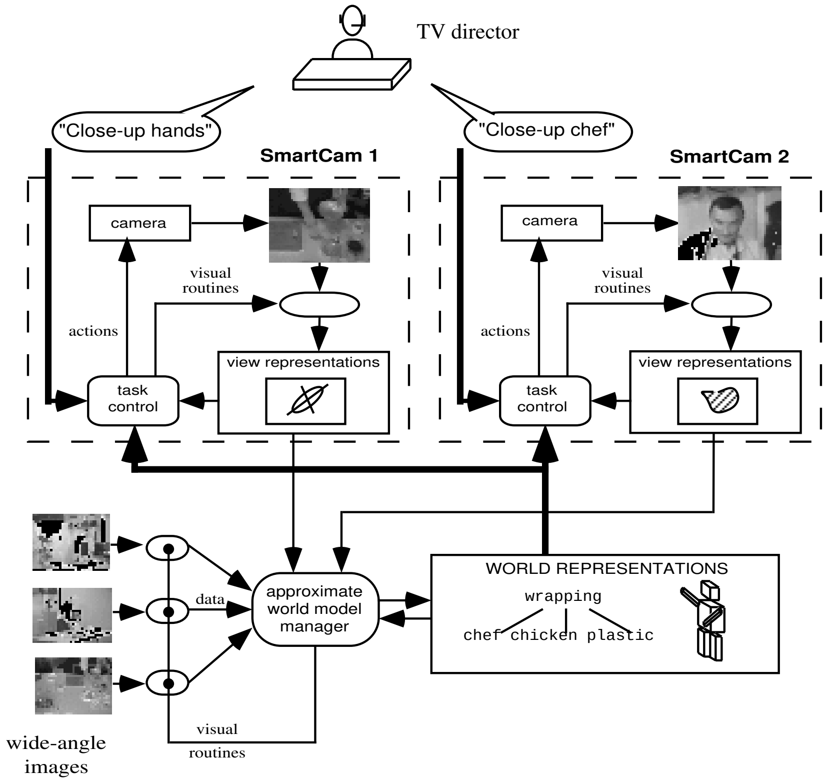

Figure 3: The architecture of a system composed by two SmartCams. The dashed squares delimit the system contained in each camera. The bottom part of the figure shows the structure of the modules responsible for maintaining the approximate world representations.

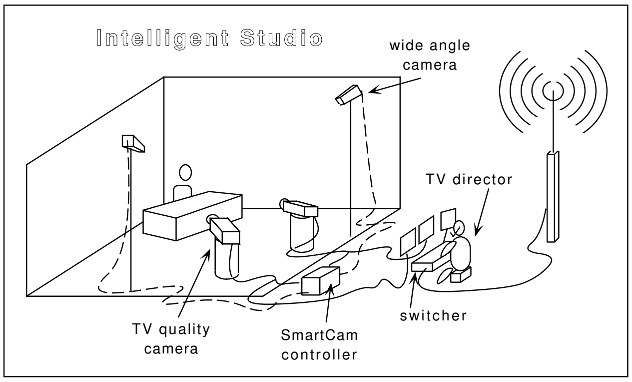

Figure 4: The concept of an intelligent studio, which uses wide-angle imagery to understand the events happening in the studio, and TV quality cameras to generate the images to be aired.

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 Figure 2. This routine detects the largest moving blob in a pair of consecutive frames, and 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 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` used as a head detector, are defined by the following rule:

```
IF the projection of the approximate model of the target is
      partially contained in the camera's current image AND
   the 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
```

The same vision routine can be applied in different situations for a variety of tasks producing outputs with different meanings. As a example, we implemented a simpler version of the same routine, called

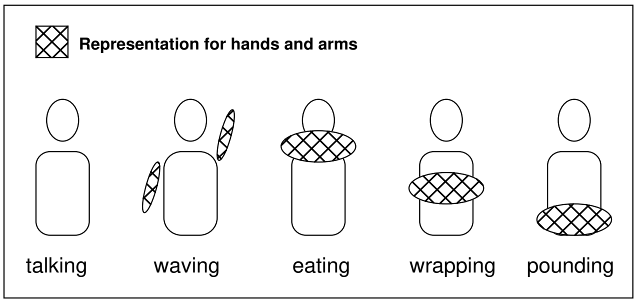

Figure 5: Different representations for the hands and arms are used according to the current action. When the subject is talking, no representation is necessary; when waving, each arm is represented separatedly, while in the case of manipulative actions the hands are represented by one big ellipsoid.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

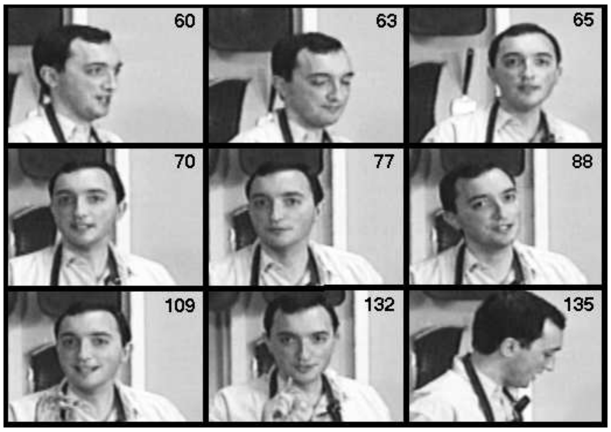

Figure 6: Response of the lateral camera to the call "`close-up chef`".

`region-moving-blob` which detects moving blobs in a pair of consecutive images. When applied to a relatively wide-angle image containing one moving person, the routine can be used to detect the area corresponding to the body of the person. The same routine, however, when applied in the context of a close-up of the hands, returns the region corresponding to the movement of the hands.

An important use of the approximate models is checking the output of the vision routines. Many times the lack of object movement makes the movement-based routines return tiny, incorrectly positioned regions. In this case the area of the returned region is compared with the area of the projected approximate model, and regions without proper values are dismissed. As mentioned before, this kind of post-condition can be considered as part of the appropriate context for a routine.

The approximate representations of actions are also employed as sources of context to non-visual routines, at different levels. The approximate world model manager 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 it is waving, each arm is also represented by two different 3D models; and if the subject is manipulating some object, then both hands are represented by an single ellipsoid corresponding to the area where the hand action is likely to occur. The expected initial position of the hands is also influenced by the action, as demonstrated by the different heights for eating, wrapping and pounding in fig. 5.

Knowing the current action changes the information required by a specific framing task. A medium close shot of a person who is manipulating objects 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.

Finally, the approximate model of the world is also used to drive the cameras when a new call is received. Often a request for framing involves an object that is not in the current field of view of the camera. Suppose that, after a close-up of the chef, a close-up of the chopping board is called. In this situation there are probably no view representations concerning the new target, and the camera must refer to the approximate models to obtain an initial clue about the position. After moving to the new location (and typically zooming out as well), the task control module of the camera can start checking the applicability of the vision routines responsible for finding view representations of the targeted object.

7 Examples of the 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. As mentioned before, the initial position of the chef and the objects, and a timed script are given. 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, enabling the correction in the framing.

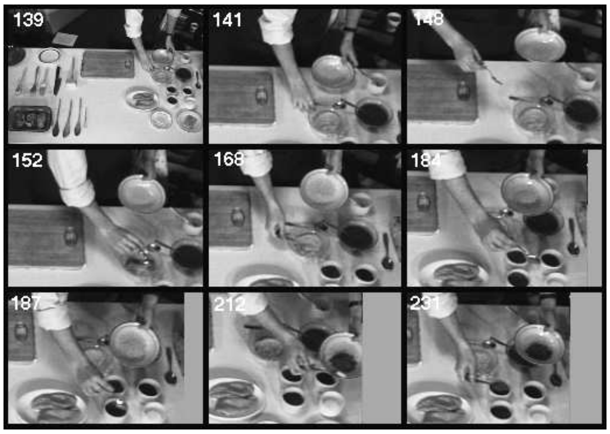

Figure 7: Response of the ceiling SmartCam to the call "`close-up hands`". The grey areas to the right of the last frames correspond to areas outside the field of view of the wide-angle image sequence used by the simulator.

Figure 8: Response of the lateral SmartCam to the call "`medium-close-shot chef`".

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 base 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 was asking 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 on the chopping board, 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 whenever the images from a camera are put "on-air".

8 Conclusion and Future Directions

We have proposed a mechanism for allowing contextual knowledge to impact the selection of view-based vision routines to perform perceptual tasks. The basic idea is to use an approximate world model to determine whether the applicability conditions of vision routines are satisfied by the current world situation.

We are employing a multi-representational system (similar to [4]) 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 ([2, 16]). We have demonstrated encouraging results in the domain of the Intelligent Studio using SmartCams to respond to a TV director's request for particular shots of a cooking show.

At present our system for control of TV cameras is still being developed. One of our major concerns 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 beginning and the end of actions, and to recognize them.

The findings of Newtson et al. ([14]) and Thibadeau ([22]) 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 temporal action segmentation is still a considerable task.

Part of the difficulties comes from the problem of recognizing actions and body movements. Some research has been done in recognizing movements as, for instance, the works of Tsotsos *et al.* ([23]) and Rohr ([19]), and on theoretical grounds by Allen ([1]), Polana and Nelson ([17]), and Israel *et al.* ([8]). Research in action recognition has been more rare (see the work of Kuniyoshi and Inoue [10]), though we believe the use of approximate models can significantly facilitate the provision of the contextual information which is essential for action recognition.

Another interesting direction 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. Prokopowicz et. al. ([18]) examines some typical characteristics of such a language. However, this work lacks representations for the objects of the world in a way which makes some of their properties derivable, instead of completely "pre-compiled", what we believe is conceptually more feasible to be implemented using our approximate world models.

Finally, algorithm's complexity is also a very relevant issue. Can the use of approximate models be justified by efficiency reasons? Although the complexity of active vs. passive visual search has been successfully addressed by Tsotsos ([24]), the analysis of the impact of approximate world models probably requires significantly more elaboration since different amounts of *a priori* information have strong influence in the final performance of the system. The analysis, thus, may require the use of methods for measuring *a priori* information: a possible direction is the use of concepts borrowed from PAC-learning theory ([12]).

References

[1] J. F. Allen, "Towards a General Theory of Action and Time," *Artificial Intelligence,* vol. 23, pp. 123–154, 1984.

[2] J. Y. Aloimonos, "Purposive and Qualitative Active Vision," *Proc. of Image Understanding Workshop,* Pittsburgh, Pennsylvania, pp. 816–828, September 1990.

[3] D. J. Beymer, "Face Recognition Under Varying Pose," *Proc. of CVPR'94,* Seattle, Washington, June 21–23, pp. 756–761, 1994.

[4] A. F. Bobick and R. C. Bolles, "Representation Space: An Approach to the Integration of Visual Information," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,* San Diego, California, June 4–8, pp. 492–499, 1989.

[5] D. D. Fu, K. J. Hammond, and M. J. Swain, "Vision and Navigation in Man-made Environments: Looking for syrup in all the right places," *Proc. of the Workshop on Visual Behaviors,* Seattle, Washington, June 19, pp. 20–26, 1994.

[6] A. R. Hanson and E. M. Riseman, "VISIONS: A computer system for interpreting scenes," *Computer Vision Systems,* Academic Press, New York, pp. 303–333, 1978.

[7] B. Horn and M. J. Brooks (eds.), *Shape from Shading,* The MIT Press, Cambridge, Massachusetts, 577 pgs, 1989.

[8] D. Israel, J. Perry, and S. Tutiya, "Actions and Movements," *12th IJCAI,* Sydney, Australia, August 24–30, pp. 1060–1065, 1991.

[9] R. C. Jain and T. O. Binford, "Ignorance, Myopia, and Naiveté in Computer Vision Systems," *CVGIP: Image Understanding,* vol. 53(1), pp. 112–117, January 1991.

[10] Y. Kuniyoshi and H. Inoue, "Qualitative Recognition of Ongoing Human Action Sequences," *Proc. of IJCAI-93,* pp. 1600–1609, 1993.

[11] D. B. Lenat and R. V. Guha, *Building Large Knowledge-Based Systems,* Addison-Wesley, 1990.

[12] Y. Mansour and R. Rivest, "Results on Learnability and the Vapnik-Chervonenkis Dimension," *Information and Computation,* vol. 90(1), pp. 33–49, January 1991.

[13] B. Moghaddam and A. Pentland, "Face Recognition using View-Based and Modular Eigenspaces," *Automatic Systems for the Identification and Inspection of Humans,* SPIE vol. 2277, July 1994.

[14] D. Newtson, G. Engquist, and J. Bois, "The Objective Basis of Behavior Units," *Journal of Personality and Social Psychology,* vol. 35(12), pp. 847–862, December 1977.

[15] J. Oliensis,"Shape from Shading as a Partially Well-Constrained Problem," *CVGIP: Image Understanding,* vol. 54(2), pp. 75–104, 1991.

[16] C. Pinhanez, "Controlling a Highly Reactive Camera Using a Subsumption Architecture," *Proc. of Applications of AI 93: Machine Vision and Robotics,* Orlando, Florida, pp. 100–111, 1993.

[17] R. Polana and R. Nelson, "Low Level Recognition of Human Motion," *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects,* Austin, Texas, November 11–12, pp. 77–82, 1994.

[18] P. N. Prokopowicz, M. J. Swain, and R. E. Kahn, "Task and Environment-Sensitive Tracking," *Proc. of the Workshop on Visual Behaviors,* Seattle, Washington, June 19. pp. 73–78, 1994.

[19] K. Rohr, "Towards Model-Based Recognition of Human Movements in Image Sequences," *CGVIP: Image Understanding,* vol. 59(1), pp. 94–15, January 1994.

[20] T. M. Strat and M. A. Fischler, "Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery," *IEEE PAMI,* vol. 13(10), pp. 1050–1065, October 1991.

[21] M. J. Tarr and M. J. Black, "A Computational and Evolutionary Perspective of the Role of Representation in Vision," *CVGIP: Image Understanding,* vol. 60(1), pp. 65–73, July 1994.

[22] R. Thibadeau, "Artificial Perception of Actions," *Cognitive Science,* vol. 10, pp. 117-149, 1986.

[23] J. K. Tsotsos, J. Mylopoulos, H. D. Covvey, and S. W. Zucker, "A Framework for Visual Motion Understanding," *IEEE-PAMI,* vol. 2(6), pp. 563–573, November 1980.

[24] J. K. Tsotsos, "On the Relative Complexity of Active vs. Passive Visual Search," *IJCV,* vol. 7(2), pp. 127–141, 1992.

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