M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 357
Submitted to CVPR'96
October 1995

Divide and Conquer: Using Approximate World Models to Control View-Based Algorithms

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

Abstract

Most view-based vision algorithms are based on strong assumptions about the disposition of the objects in the image. To safely apply those algorithms in real world image sequences, we are proposing that a vision system should be divided into two components. The first component contains an *approximate world model* of the scene — a low accuracy, coarse description of the objects and actions in the world. Approximate world models are constructed and updated by simple vision routines and by the use of contextual information. The second component employs view-based algorithms to perform required perceptual tasks; the selection and control of the view-based methods are determined by the information provided by the approximate world model. We demonstrate the approximate world model approach 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.

1 Introduction

Many of the methods developed by computer vision research are dependent on strong assumptions about the objects and the actions depicted in the image. Among them, the so called *view-based* methods rely on the specific disposition of the objects in the view ([2, 7]). However, most of the interesting applications of computer vision occur in situations where the disposition of the objects in the view is unknown and/or changes with time, forcing the use of view-independent methods which are often computationally very expensive.

We propose that vision systems should maintain a simple model of the scene containing the information needed to dynamically select and control view-based vision routines. The basic idea is that the vision system constructs an **approximate world model** of the scene using simple, low accuracy vision methods and contextual information. The approximate world model may include geometrical models, categorical spatial information, as well as general action and event specifications. View-based routines are then encapsulated in *applicability rules* which specify the conditions when a routine can be safely applied to imagery. The information used to check the conditions is obtained from the approximate model.

Essentially, we are proposing a "divide and conquer" approach to the construction of vision systems, where the task is divided into two components: the first deals with the complexity of the objects and their interactions in the real world, sacrificing accuracy if necessary. The second component contains view-based algorithms with enough accuracy to satisfy the task requirements, though sensitive to particular situations in the real world. In the long run, 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.

In the remaining sections we discuss more fully the type of approximate models we propose, and describe how those models are used in the selection of vision routines. We then describe an example domain and task — the *Intelligent Studio* — where approximate world models have been employed to construct TV cameras capable of responding to verbal requests such as "`Camera 1, give me a close-up of the chef.`"

2 Approximate Models

Many real-world vision problems are hard not because the task is complex itself, but mostly because they have to cope simultaneously with two levels of complexity: the complexity of the environment, in terms of the interactions among objects in the real world; and the complexity of a specific visual task.

For example, consider a face detection/recognition system to be used to monitor the entrance of a corporate building, recognizing the employees which walk through a gate. Even if we succeed in building a view-based method which detects faces all over the image, with any direction of sight ([3, 12]), our real system still has to deal with the problem of occlusion. On the other hand, there is no need to perform full 3-D reconstruction, or determining precisely the 3-D structure of every person who walks by, since the objective of the system is just to recognize people.

The fundamental point is that the real world does not need to be fully and accurately understood to detect many situations where a specific vision method is likely to fail. In the above example, if we employ a vision sub-system which only determines the position of the center of a person's head within a 2 foot error range, such system is still able to detect many situations where the face recognizer is likely to fail, and, possibly, even speed up the face detection/recognition algorithm by providing an initial cue of the head position.

In many environments, coarse reconstruction of the 3-D world is within the grasp of current computer vision capabilities. In our gate monitoring example, it seems plausible to construct a vision system that monitors the flow of people, and determines the position and attitude of them with relatively low accuracy (using techniques as those described in [18, 20]). And, as computer vision progresses, more domains are likely to be suitable for the computation of low accuracy representations.

These coarse descriptions of the main elements of a scene are called *approximate models*. Our emphasis on the approximate nature of these models is related to the intended use of this information: approximate models are not to be exploited directly by vision-guided task modules, but to be used internally by a vision system in the selection and control of vision routines. Therefore, the term "approximate" is understood here as imprecise, perhaps inadequate in terms of being sufficient to accomplish a given task; the competence required is that the model encode enough information to be able to guide the selection of appropriate vision routines.

World Models

A common reason for the failure of some view-based routines is related to the 3-dimensional nature of the world, as, for example, when template matching routines produce wrong results due to partial occlusion, change of attitude of the objects, or shadows. Therefore, to use an approximate model to control the application of such routines, it is necessary that the approximate models represent some of the 3-D information of the scene. In those cases, the simplest approach may be representing the scene independently from any particular point of view, and to include information about the cameras' attitude and position so that the system is able to compute the likely view from each camera. We term such view-independent 3-D descriptions of the world as *approximate world models*.

Maintaining an approximate world model requires additional sensing and computation which might not be required to directly address current perceptual tasks. We must be willing to incur this additional cost to increase the competence of the view-based routines. However, the computation of approximate models can exploit common forms of contextual or semantic information. Contextual and semantic information have not been extensively in computer vision because of their inability to provide accurate data. If the accuracy requirements are relaxed, as it is in the case of approximate world models, context can be used to predict possible positions and attitudes of objects as we will show in a later section (see also [5]).

Previous Work

Coarse and/or hierarchical descriptions have been used before in computer vision ([4, 11]). Particularly, Bobick and Bolles ([4]) employed a multi-level representational system where different queries were answered by different representations of the same object. Part of the novelty of our work is related to the use of the models in the dynamic selection of appropriate view-based methods according to the world situation.

Moreover, the basic aim of our approximate world models is to cope with the complexity of a real world environment, easing the burden on the vision routines responsible to perform the actual visual task. Comparing to related works, like Strat and Fischler's *Condor* system ([19]), approximate world models provide a much more clear distinction between the view-based component and the 3-D world component, enabling more modular systems where new view-based routines can be incorporated easily as long as they require similar information from the approximate world model.

Reconstructionist vs. Purposive Vision

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). Our proposal falls between the strict reconstructionist and purely purposive strategies. We are arguing that reconstruction should exist at the approximate level, guiding the purposive vision routines of the view-based level. 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.

3 Controlling View-Based Routines

Our goal is to use approximate world models in the control of view-based vision routines, since the performance of those routines critically depends on appropriate viewing conditions. 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. Approximate world models allow the prediction of the disposition and size of the many elements in a view, which is a fundamental part of the applicability conditions in the case of view-based vision routines.

In order to use a vision routine only in situations where its applicability conditions are valid, we encapsulate each vision routine in an *applicability rule*, which describes pre-conditions, application constraints, and post-conditions in terms of general properties about the targeted object, other objects in the scene, the camera's point of view, and the result of the vision routine. Given the state of the world, as described by the approximate model, the job of the vision system is to verify which applicability rules have their conditions satisfied, and then to appropriately apply the vision routines which pass the test.

Consider the example of the vision routine "`extract-moving-blob`" designed to detect a moving region in a sequence of two consecutive frames using simple frame differencing. We can describe a possible set of applicability conditions for the routine

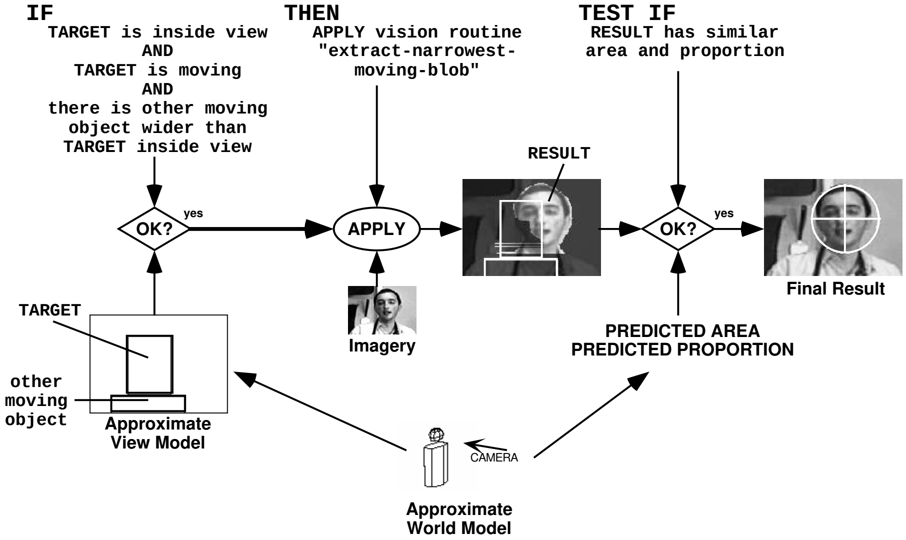

Figure 1: Example of an applicability rule for a vision routine and how the information from the approximate world model is used.

"`extract-moving-blob`" by the following applicability rule:

```
IF
   TARGET is inside view AND
   TARGET occupies a reasonable portion
     of the view AND
   TARGET can move AND
   TARGET is not occluded by other object AND
   TARGET is not in front of another moving
     object whose region contains TARGET
THEN
   apply extract-moving-blob to a region
     around the bounding box of TARGET,
     producing RESULT
TEST IF
   RESULT has similar area and
     proportion as TARGET AND
   RESULT center is close to TARGET center
```

To use such a rule, the vision system must maintain an approximate model of `TARGET` which is rich enough to provide the *approximate information* required by the rule, such as estimations of the target's center, area, bounding box, and depth. Moreover, the system must also have some information about potential distractors, such as large moving objects behind the target.

Certainly "`extract-moving-blob`", with its simple algorithm, assumes many more constraints on the input images. However, the use of post-conditions (`TEST IF`) reduces the possibility that an incomplete specification of pre-conditions generates incorrect results. For instance, in the case of "`extract-moving-blob`", often the lack of actual object movement makes the routine return tiny, incorrectly positioned regions which are filtered out by the post-conditions.

It is important to notice that by defining the pre- and post-conditions in terms of a generic object (`TARGET`) and generic attributes such as area and bounding box, "`extract-moving-blob`" can be applied in different situations, for different targets, producing outputs with different meanings. As an example, this routine can be used both to detect a moving person in a relatively wide-angle image of a scene, or to find a head in a close-up view.

Figure 1 depicts the applicability rule of another vision routine, "`extract-narrowest-moving-blob`", and how the rule is applied. The routine "`extract-narrowest-moving-blob`" is a more specific version of "`extract-moving-blob`", where the resulting moving blob is divided into two areas, of which the narrowest is returned. The routine can be used to detect heads in close-ups as shown in fig. 1. In the figure, the geometric 3-D approximate models of the head and the trunk enable the vision system to obtain an *approximate view model*, which assures that two moving areas with different widths are expected to be present in the image.

Figure 1 also shows a real situation where, in spite of the inaccurate predictions of the position and size of the head and the trunk, "`extract-narrowest-moving-blob`" is able to detect the head in its right position: the rectangle representing the predicted position of the head (according to the approximate model) is quite misaligned while the recovered area, labeled `RESULT`, covers the head correctly.

It is important to differentiate the concept of applicability rules from rule-based or expert-system approaches to computer vision ([6, 23]). Although we use the same keywords (`IF`, `THEN`), the implied control structure has no resemblance to a traditional rule-based system: there is no inference or chaining of results. Expert systems are normally built around the assumption that the final answer is eventually reached by growing facts in the blackboard area. However, most view-based methods are designed to provide directly the necessary information required by a perceptual task. Therefore, our simple control mechanism, though incapable of inference and chaining, seems to be sufficiently powerful.

4 An Implementation: 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*, and their task is to provide the correct framing based upon the director's calls and the action taking place.

Framing is more difficult than object localization and requires more information. For instance, a call for a close-up of a subject 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 (see [25], pp. 111–122, for simple rules of framing). Framing also requires knowledge about the current actions; we illustrate this in a later example.

The basic architecture of a SmartCam is shown in fig. 2. There are two basic modules: the first performs view-based tracking and framing, and the second maintains an approximate world model of the TV studio. The view-based module (surrounded by a dashed square) works in a feedback loop fed both by requests from the TV director and by information from the approximate world model. 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 representations which are view-dependent.

Considering the requirements of this application, the approximate world model is designed to represent the subjects and objects in the scene as 3-D blocks, cylinders, and ellipsoids, and to use symbolic representations (slots and keywords) for features like direction of sight and actions. To each object — and its different representations — corresponds an individual name, which is also used by the TV director when communicating with the SmartCams.

The 3-D representations are positioned in a 3-D virtual space corresponding to the TV studio. The cameras' calibration parameters area also approximately known, enabling the system to calculate the projection of the objects into the camera plane. The precision can be quite low, and in our system the position of an object might be off by an amount comparable to its size.

The view-based tracking algorithms are designed to use information with this level of uncertainty when determining the position of an object (on the image plane) precisely enough for the framing task. Presently almost all vision routines are based upon motion-detection — similar to the "`extract-moving-blob`" routine mentioned before.

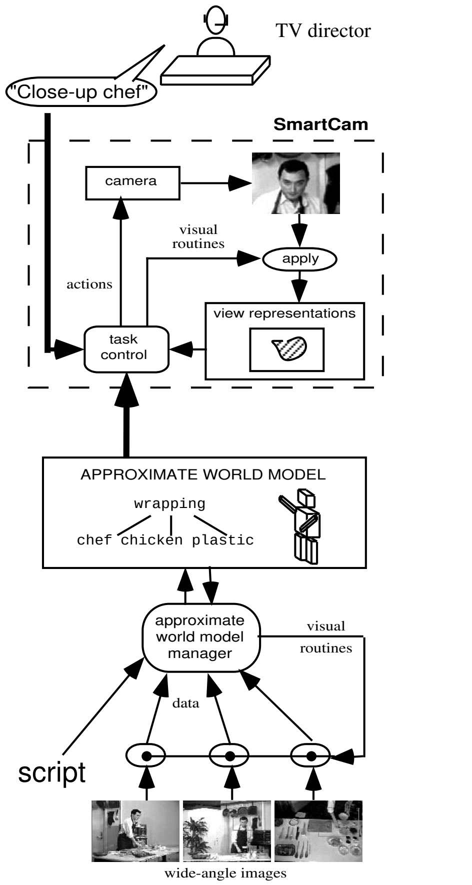

Figure 2: The architecture of a SmartCam. The dashed square delimits the system contained in a particular camera. The bottom part of the figure shows the structure of the modules responsible for maintaining the approximate world models.

The applicability rules are able to prevent the use of such simple methods in most of the unsafe situations, and especially, to filter out errors by checking the results against the predictions based on the approximate world models.

The approximate world model also contains symbolic descriptions of the objects and actions in the scene. The symbolic description of an object includes information about to which class of objects the object belongs, its potential use, and its roles in the current actions. For instance, a bowl is a member of the class of solid objects which can be manipulated, and its representation describes the subject who is handling the bowl if that is the current case.

During the initialization process, the 3-D representations of the objects are given manually, describing the initial shape and position of all objects. This information is provided only for the first frame of the sequence. Also available is a script that mimics standard TV scripts, *i.e.*, a sequence of short descriptions of the actions which are going to happen in the studio. For instance, in a cooking show, the script can contain a sequence like: "`Chef talks about today's` recipe. Chef turns to center camera. Chef mixes ingredients in a bowl".

In order to keep the approximate world model current, and especially its 3-D representations, a SmartCam employs wide-angle cameras to monitor the studio. The images from these cameras are processed using vision routines (again based on motion) able to detect movements of the main components of the scene. The motion components of an object detected by the different wide-angle cameras are integrated to determine the movement of the object in the 3-D world.

The information in the script is especially helpful in the detection of drastic changes in the states of the objects in the scene. For example, if it is known from the script that a bowl is being used by a chef, the system can position the 3-D model of the bowl near the position of the hands (for more details, see [15]). The approximate world manager also updates the corresponding states of the symbolic representation of the bowl.

Information about the size and position of objects, available from the approximate world model, also enables our vision system to deal more easily with adaptive parameters. For instance, the motion-detection routines are controlled by 3 different parameters: number of frames between the two subtracted frames, blurring level, and a threshold for binarization. If the region returned by the routine is too small compared to the area predicted by the approximate world model, the task control module makes new attempts using first a smaller binary threshold, then more blurring, and finally a longer time span. Similar procedures adjust the parameters when the returned area is too large. The results are always checked, and successful parameters are saved to be used as initial parameters in the processing of the next frame.

In situations where multiple SmartCams are framing the same scene, they can share the information from the same approximate world model, creating an *Intelligent Studio* ([14], and see fig. 3). In an Intelligent Studio, coarse, fixed, wide-angle cameras are used to maintain the approximate world models, while high-quality mobile cameras provide the images requested by the TV director, using when necessary the information from the common approximate world model.

5 Examples of the Results

The system which produced the results shown in this paper does not use real moving cameras, but simulates them using a moving window on wide-angle images of the set. Several examples of a 5-minute scene as viewed by three wide-angle cameras were recorded and digitized. The SmartCam output image is simulated by extracting a rectangular window of some size from the wide-angle images.

A "cooking show" is the first domain in which we are experimenting with our SmartCams. Part of a typical script of a cooking show is shown in fig. 4 in the same format as it is received by the approximate world manager. One important difference between the script of fig. 4 and a realistic script is that the start and finish of each action segment are given explicitly.

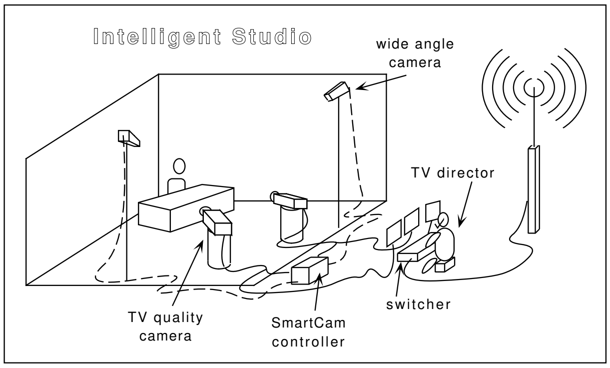

Figure 3: The concept of an Intelligent Studio, which uses wide-angle imagery to understand the events happening in the studio, and TV quality cameras to generate the images to be aired.

```
                    ⋮
(From time 0:13s to 0:26s)
    Chef talks about today's recipe
    (to side camera).
(From time 0:26s to 0:27s)
    Chef turns to center camera.
(From time 0:27s to 0:55s)
    Chef mixes paprika, basil, bread
    crumbs, and salt in a bowl.
(From time 0:55s to 0:67s)
    Chef wraps chicken with a plastic bag.
                    ⋮
```

Figure 4: Part of a typical cooking show script, but including the time references needed in the current version of the SmartCams.

Currently, the times are obtained manually from the pre-recorded sequence; of course this is feasible only in the simulated version of the SmartCams we are experimenting with. In fact, the focus of our current research is precisely the visual determination of the onset of an anticipated action, according to the sequence in the script.

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. All the results obtained so far employ only very simple vision routines similar to "`extract-moving-blob`" and "`extract-narrowest-moving-blob`", based on movement detection by frame differencing.

In fig. 5 we can see that the inaccuracy of the approximate world models do not affect the final results of the vision routines. Two SmartCams, `side` and `center`, were tasked to provide a close-up of the chef. The geometric model corresponding to the chef is quite misaligned, as can be seen by its projection

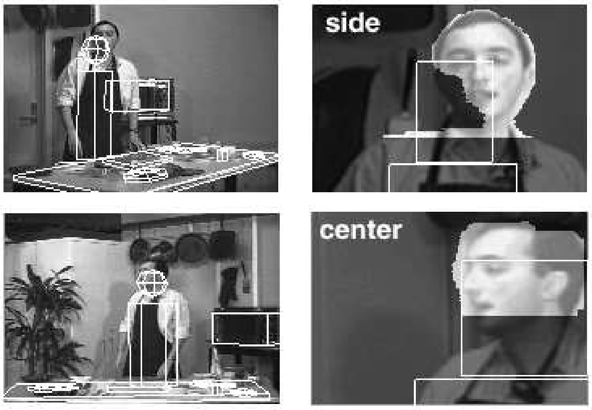

Figure 5: Example of response to the call "`close-up chef`" by two different cameras, `side` and `center`. The left images show the projection of the approximate models on the wide-angle images. The right images display the result of vision routines as highlighted regions, compared to the predicted position according to the approximate model of the head and the trunk, shown as rectangles.

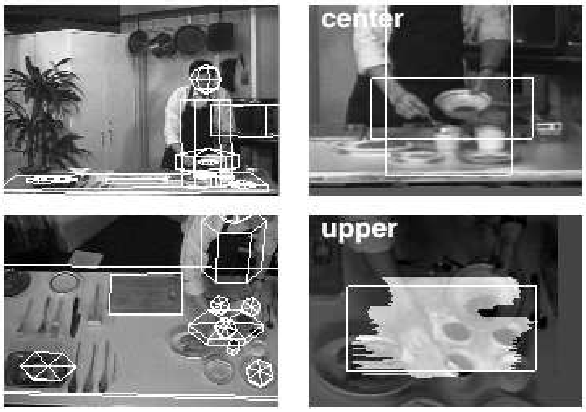

Figure 6: Example of response to the call "`close-up hands`" by two different cameras, `center` and `upper`. The left images show the projection of the approximate models on the wide-angle images. The right image of the `center` camera displays only the rectangles related to the area of the hands and the trunk. In this case the hands could not be detected, since there is a moving region (the trunk) situated behind the hands, precluding the call of any movement-based vision routine (refer to the last pre-condition of "`extract-moving-blob`"). However, in the case of the `upper` camera the movement-based routines can be safely used, and the result is the highlighted region on the bottom-right image.

into the wide-angle images of the scene (left side), and in the images seen by the SmartCams, where the projections of the approximate geometric models of the head and the trunk are represented by rectangles. However, the current view of `side` satisfies the applicability conditions of a routine similar to "`extract-narrowest-moving-blob`" which correctly detects the position of the head (top-right of fig. 5). In the case of the `center` camera, a routine similar to "`extract-moving-blob`" is applicable, and generates the highlighted area in the bottom-right image of fig. 5.

Figure 6 shows a situation where the approximate world models avoid the application of a routine in an unsafe condition. In this case, both cameras were asked to provide a close-up of the hands. In this situation both hands are approximately modeled by an ellipsoid in front of the trunk. This model comes from the information contained in the script indicating that a "`mix`" action is happening at the current frame. Based solely on knowing the current action, it is possible to predict that the hands are likely to be in the front on the trunk, at height of the waist.

In those conditions a routine similar to "`extract-moving-blob`" can be applied to the `upper` camera images, resulting in the highlighted area on the bottom-right of fig. 6. However, in the case of the `center` camera, the predicted region for the hands is in front of the region of the trunk (see top-right image of fig. 6). Therefore a simple routine like "`extract-moving-blob`" should not be applied, in order to avoid detecting the movement of a background object (the trunk) instead of the movement of the hands. If all other applicability rules also fail, the system knows that it can not robustly determine the position of the hands from that particular viewpoint.

Figure 7 shows typical framing results obtained by the system. The leftmost column of fig. 7 displays some frames generated in response to the call "`close-up chef`". The center column of fig. 7 contains another sequence of frames, showing the images provided by a SmartCam tasked to provide "`close-up hands`". This sequence contains a great deal of small corrections because the chef is taking ingredients from bowls and cups situated in different positions on the table.

The rightmost column of fig. 7 is the response to a call for a "`medium-close-shot 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 maintains this constraint as much as possible. 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 same WWW-site contains the results from the processing of a similar sequence, containing another performance of the same script. In this case the chef is wearing glasses, and the actions are performed in a faster speed. The vision system in both cases is exactly the same, but the time references in the script are changed.

Figure 7: Responses to the calls "`close-up chef`", "`close-up hands`", and "`medium-close-shot chef`". Refer to background objects to verify the amount of correction needed to answer those calls appropriately. The grey areas to the right of the last frames of the sequence "`close-up hands`" correspond to areas outside the field of view of the wide-angle image sequence used by the simulator.

To evaluate better the robustness of the adaptive parameter scheme described earlier, we also experimented processing a 10:1 jpeg-compressed version of the input sequences. In spite of all the artifacts in the images, the SmartCams were basically able to repeat the performance previously achieved, though using very different sets of parameters.

6 Conclusions

We have proposed that vision systems should maintain a simple model of the scene containing the information needed to dynamically select and control view-based vision routines. 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 have demonstrated encouraging results in the domain of the Intelligent Studio, using SmartCams to provide shots of a cooking show.

To date, approximate world models have been employed only in the control of SmartCams. However, there are many other real domains where vision systems can exploit the simplicity afforded by the use of approximate world models. For instance, consider the example of monitoring a parking lot: a system could be designed based on template matching routines, and an approximate world model could be used to avoid their use in situations of likely occlusion. Another interesting example is the face recognizer/detector describe earlier.

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 to use scripts describing only the likely sequence of events. To use such scripts, the system must be able to visually recognize actions, or, at least to identify the beginning and the end of actions if assuming that the sequence of actions is correct.

In this last case, there is experimental evidence 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 segmentation processes ([13, 22]). Thus, a non-timed version of the script would, theoretically, give most of the information needed.

Some research has been done in recognizing human movements [24, 18, 1, 16, 8] and in action recognition [10], though most methods were developed for situations much more constrained than those found in normal TV studios. However, we believe that 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. [17] examines some typical characteristics of such a language. However, their work lacks representations for the objects of the world that allows the derivation of view-dependent properties. We believe that our approximate world models provide a more adequate framework to supply the essential view-dependent information.

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, "The Representation Space Paradigm of Concurrent Evolving Object Descriptions," *IEEE PAMI,* vol. 14(2), pp. 146–156, January 1992.

[5] A. F. Bobick and C. Pinhanez, "Using Approximate Models as Source of Contextual Information for Vision Processing," in *Proc. of the ICCV'95 Workshop on Context-Based Vision,* Cambridge, Massachusetts, pp. 13–21, June 1995.

[6] B. A. Draper, R. T. Collins, J. Brolio, J. Griffith, A. R. Hanson, E. M. Riseman, "Tools and Experiments in the Knowledge-Directed Interpretation of Road Scenes", in *Proc. of the DARPA Image Understanding Workshop*, Los Angeles, California, pp. 178–193, February 1987.

[7] C. Fermüller, "Global 3-D Motion Estimation," *Proc. of CVPR'93,* New York City, New York, June 15–17, pp. 415–421, 1993.

[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. Marr and H. K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," in *Proc. R. Soc. Lond. B*, vol. 200, pp. 269–294, 1978.

[12] 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.

[13] 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.

[14] C. Pinhanez and A. F. Bobick, "Intelligent Studios: Using Computer Vision to Control TV Cameras," *Proc. of IJCAI'95 Workshop on Entertainment and AI/Alife,* Montreal, Canada, pp. 69–76, August 1995.

[15] C. Pinhanez and A. F. Bobick, "Scripts in Machine Understanding of Image Sequences," to appear in *Proc. of AAAI Fall Symposium on Computational Models for Integrating Language and Vision,* Cambridge, Massachusetts, November 1995.

[16] 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.

[17] 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.

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

[19] 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.

[20] G. D. Sullivan, A. D. Worrall, and J. M. Ferryman, "Visual Object Recognition Using Deformable Models of Vehicles," in *Proc. of the ICCV'95 Workshop on Context-Based Vision,* Cambridge, Massachusetts, pp. 75–86, June 1995.

[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, "Knowledge Organization and its Role in Representation and Interpretation of Time-Varying Data: The ALVEN System," *Computational Intelligence,* vol. 1, pp. 16–32, 1985.

[24] 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.

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