M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 489
April 1999. Appears in Proceedings of the National Conference on Artificial Intelligence (AAAI), July, 1999. (shortened version in CVPR '99).

A Framework for Recognizing Multi-Agent Action from Visual Evidence

Stephen S. Intille and Aaron F. Bobick
Perceptual Computing Group, MIT Media Laboratory
20 Ames St., Cambridge, MA 02139
(intille | bobick @media.mit.edu)

Abstract

A probabilistic framework for representing and visually recognizing complex multi-agent action is presented. Motivated by work in model-based object recognition and designed for the recognition of action from *visual evidence*, the representation has three components: (1) temporal structure descriptions representing the temporal relationships between agent goals, (2) belief networks for probabilistically representing and recognizing individual agent goals from visual evidence, and (3) belief networks automatically generated from the temporal structure descriptions that support the recognition of the complex action. We describe our current work on recognizing American football plays from noisy trajectory data.[1]

Keywords: action recognition, plan recognition, representing visual uncertainty

1 Introduction

Evaluating whether an observed set of visual phenomena constitute a particular dynamic event requires representation and recognition of temporal relationships and uncertain information. The goal of this paper is to present a new approach to the representation and recognition of complex multi-agent probabilistic actions. By *complex* we simply mean that the action contains many components that occur in, typically, a partially ordered temporal relation to one another, subject to certain logical constraints (e.g. `A` happens `before` `B`, `B` is `before` `C` or `D`, but only one of `C` or `D` can occur). These relations generally reflect causal connections or influences between components. The actions we are considering are *multi-agent*, resulting in parallel event streams that interact in interesting temporal (typically causal) ways.

By *probabilistic* we refer to the uncertain nature of both the model and the data. The action description itself is typically probabilistic: e.g. `B` follows `A`, but only 80% of the time. This uncertainty results from complex actions defined by typical components that are only sometimes observed due to uncertainty in the world. Another source of uncertainty is the fuzziness of attributes used to describe agent interaction (e.g. `obj1` is `near` `obj2`). Finally, the design of the representation is intended to support recognition and we therefore need to consider real sensing capabilities, which are probabilistic at best. Often, perceptual evidence can be either missed or hallucinated.

There are numerous domains that contain interesting, complex, probabilistic actions. Examples include sporting events, military and security surveillance, traffic monitoring, and robotic collaboration. The task and domain developed here is recognizing American football plays. It has the necessary attributes of containing complex actions (plays) performed by a multi-agent system (the offense) in which there is great uncertainty and unpredictability (the defense). Methods exist for tracking football players from video [12]. For the recognition task, we presume tracked data that provides the location and rough orientation of each player at each time during the play. Our current system uses a database of 29 manually, though noisily, tracked plays. Figure 1 shows 3 "chalkboard" image examples of 3 different observations of a "p51curl" play.

1.1 An analogy to object recognition

At the heart of our approach to complex action recognition is an idea developed within the context of model-based object recognition. The task there is to match a given object model to an image from which edge elements have been extracted. One of the more successful approaches to this problem is that of using feature-model interpretation matching trees, where the visual features are edge segments [8]. Each layer of the tree represents a given model edge. The fanouts of each node span the potential image edge fragments that might match the given model edge of the given layer. A *hypothesis* is a path from the root to the leaves that specifies the match of each model edge to specific image features.

The goal, of course, is to find the correct hypotheses. However the number of edges make exhaustive search computationally prohibitive. Rather, the approach is to find a *consistent* hypothesis, and assume that consistency implies correctness. As developed in [8] the *order of the consistency* can be varied depending upon computational resources and accuracy requirements. For example, if we restrict our attention to two-dimensional objects, a unary consistency check simply requires that each model edge is at least as long as the proposed matching image edge. A binary consistency check verifies not only the unary relations but also all pairwise relationships, namely the angle and bounded distance between edges.

Grimson and Lozano-Pérez [8] note that although it is mathematically possible for an incorrect interpretation to satisfy the binary relations but not higher order relations, the probability of an object doing so falls precipitously as object complexity increases. This allows them to construct heuristic pruning methods that search for the correct interpretation by only maintaining binary consistency. It is this idea, *that massive low order consistency typically implies correctness,* that drives our approach to recognizing complex actions.

[1] This research was funded by Office of Research and Development (ORD) contracts 94-F133400-000 and 97-F157800-000.

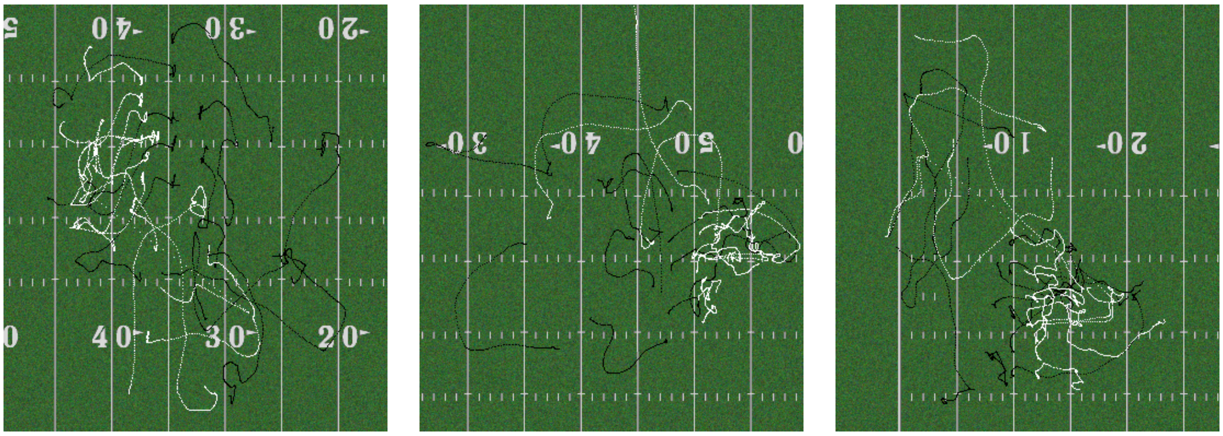

Figure 1: Three examples of a p51curl play. The lighter trajectories are the offensive players. The data provided to the system consists of trajectories for all the objects including the ball, the approximate orientation of each object at each point along its trajectory, and a position label for each trajectory.

1.2 Our approach

The approach we have developed consists of the following representational elements:

- We first define a *temporal structure description* of the global behavior, in this case a football play. The basic elements of this structure represent individual, local goals or events that must be detected. The relations coded in the structure are temporal constraints to be verified.
- For each basic element of the temporal structure, we define a *visual network* that detects the occurrence of the individual goal or event at a given time accounting for uncertain information.
- Temporal analysis functions are defined which evaluate the validity of a particular temporal relationships, such as `before`.
- A large multi-agent belief network is automatically constructed reflecting the temporal structure of the action. This network, similar in structure to a naive Bayesian classifier, represents a particular play using *only* beliefs and evidence about the expected temporal relationships between agent goals.

The likelihood that a particular play has been observed is computed by evaluating the appropriate belief networks.

1.3 s51 play example

The task for a recognition system is to recognize whether a given set of trajectory inputs like those illustrated by Figure 1 corresponds to a particular type of play, such as the p51curl. Normally plays consist of 11 offensive players. A simplified example of a p51curl play, called the "s51," containing only 4 offensive players and a reduced number of actions per player will be used for illustration in this paper. The s51 chalkboard diagram is shown in Figure 2.

The input to the system consists of trajectories given by (x,y,orientation,label) tuples as a function of the frame number, i.e. time. Here, *orientation* denotes the approximate upper-body orientation of the player and *label* is the name of the player's starting position.

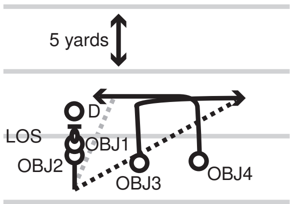

Figure 2: An football play diagramming the s51 example play. The play consists of 4 offensive agents and a ball. Also marked is the line-of-scrimmage (LOS) and some 5-yard marker yardlines. The heavy dotted line indicates the most typical path for the ball when it is thrown by OBJ2 after the ball is handed to OBJ2 from OBJ1. The lighter dotted line indicates a secondary pass option. Implicit is that OBJ3 and OBJ4 turn at the same time.

2 Prior work

Prior multi-agent plan recognition work can be roughly divided into two methods. Some approaches have an explicit representation for group intentionality (e.g. [9]), typically using modal logics. Other approaches "compile down" intentional reasoning into procedural components, trading off the ability to reason about complex intentional interaction for computational tractability in domains with noisy evidence detectors. Our hypothesis is that for some useful recognition tasks visually-detected agent-based goals can be "compiled" into efficient and powerful classifier networks using binary temporal relationships between detected goals.

Promising work on recognizing single-agent action from trajectory information using transition diagrams and fuzzy reasoning [14] led us to investigate the use of belief networks for multi-agent action recognition, which more explicitly represent knowledge dependencies and are computationally well-understood. Bayesian networks have been used to relax the strict assumptions of plan hierarchy models such as [13]. For example, networks can represent multiple top-level goals where probabilistic priors can be used to rank two equally possible but not equally likely plans [4]. Further, they have been used to integrate "action patterns" and

```
(goalTeam s51
 "Team goal for simple-p51curl (s51) play."

  (agentGoal obj1
     (agent (obj1 (C))) ; Obj1 is always the Center (C)
     (goal obj1_act1 "snapToQB (obj1)")
     (goal obj2_act2 "blockQBPass (obj1)")
     (before obj1_act1 obj1_act2))

  (agentGoal obj2
     (agent (obj2 (QB))) ;Obj2 is always the Quarterback (QB)
     (goal obj1_act1 "dropback (obj2 5)")
     (goal obj2_act2 "throwPass (obj2)")
     (before obj2_act1 obj2_act2))

  (agentGoal obj3 ;The Right Wing Back (RWB)
     (agent (obj3 (RWB RTE RHB HB FB TB LWB LSB)))
     (goal obj3_act1 "passPatStreaking
                       (obj3 4 45 defReg nearRightSidelineReg 0)")
     (goal obj3_act2 "passPatCutting
                       (obj3 70 offSidelineRightReg freeBlockingZoneReg)")
     (goal obj3_act3 "runbehind (obj3 obj4)")
     (goal obj3_act4 "passPatParaLos
                       (obj3 3 defReg offSidelineRightReg 4)")
     (goal obj3_act5 "catchPass (obj3)")
     (before obj3_act1 obj3_act2)
     (before obj3_act2 obj3_act4))

  (agentGoal obj4 ;The Right Flanker (RFL)
     (agent (obj4 (RFL RWB RSB LFL LSB LWB)))
     (goal obj4_act1 "passPatStreaking
                       (obj4 4 50 defReg offEndZoneReg 0)")
     (goal obj4_act2 "passPatCutting
                       (obj4 70 offSidelineLeftReg freeBlockingZoneReg)")
     (goal obj4_act3 "passPatParaLos
                       (obj4 3 defReg offCenterLineReg 4)")
     (goal obj4_act4 "catchPass (obj4)")
     (before obj4_act1 obj4_act2)
     (before obj4_act2 obj4_act3))

  (around obj3_act2 obj4_act2)
  (xor obj3_act5 obj4_act4))
```

Figure 3: A temporal structure description for the *s51* play example with only some actions and temporal relationships specified.

beliefs about an agent's mental state [16]. Previous work in traffic understanding has used an agent-based belief network and agent-centered features for recognition of driving activity from simulated [6] and real data [3, 10]. Unlike that work our task requires that the system must also represent the logical and temporal relationships between multiple agents. Remagnino, Tan, and Baker [17] recently described a pedestrian and car tracking and surveillance system that models the interaction between any two agents using a small belief network. Dynamic belief networks (DBNs) and hidden Markov models (HMMs) have been used with some success but have not been demonstrated to be appropriate for domains in which multi-agent relationships result in large feature spaces and in which large and complete data sets for training are unavailable.

Although some search-based systems for recognizing multi-agent goals and actions have been proposed [18, 2, 19], noisy visual data requires a representation that can handle uncertainty. [5] have demonstrated that pairwise comparison of features between trajectories can be used to recognize some group military behaviors for large numbers of agents.

Huber has shown that simple goal recognition belief networks can be constructed automatically from representations of action used for a plan generation system and then used by a planning agent in a multi-object scene [11]. Our approach builds on Huber's work of automatic construction of networks.

The remaining sections of this paper describe each component of our representation and some recognition results.

3 Temporal structure description

The temporal structure description represents the prototypical scenario of the described action. It is comprised of fundamental behavior elements connected by temporal constraints. We assume that the complex actions we wish to recognize have such a prototype and that they can be expressed with this language.

3.1 Individual goals and behaviors

We use individual agent *goals* as the basis for the descriptive structure and view complex actions as a partially ordered set of goal directed behaviors on the part of interacting agents. We *define* goals by their (probabilistic) characteristic behaviors, building on work in probabilistic plan recognition [4]. To evaluate whether an agent has a particular goal at a particular time we will evaluate the perceptual evidence.

For example, the halfback can have the goal of running between the tackle and the guard. To determine if indeed he has such a goal a recognition system must evaluate the visual evidence, particularly the position of the tackle and the guard and the direction of motion of the halfback. The interaction of multiple agents and the reaction of agents to the movement of other agents can lead to large variations in some movement, as indicated by the examples in Figure 1. However, at any given time, evidence detected in a local space-time window can indicate that an agent has a particular goal. Later we will more fully detail the construction of belief networks that serve as the definition of the individual agent goals.

3.2 Goal action components

Figure 3 shows a simplified temporal structure description for the s51 example in Figure 2. The description contains four agents: obj1, obj2, obj3, and obj4. Each object in the temporal structure graph has a set of goal action components. The example indicates that in an s51 play, obj1 should have a goal to snapToQB (snap (or hand) the ball to the quarterback) and blockQBPass (block for the QB as the QB passes the ball). Each goal has a label, such as obj1_act1 (short for object1's action1). The s51 example has been limited to just six goal types: snapToQB, blockQBPass, passPatStreaking, passPatCutting, passPatParaLos, and catchPass. The detector for each goal type receives a list of parameters.[2]

3.3 Object assignment

The trajectories in our dataset are labeled using standard football position notations (e.g. QB, C, HB). However, since all football plays can be run from several different starting formations (so that the defense cannot determine the play from the starting formation of the offense), the temporal structure description must indicate the valid position types for each object. In the example description in Figure 3, the `agent` slot of the `agentGoal obj3` description indicates that object `obj3` can possibly match with a trajectory if the trajectory has one of labels `(RWB RTE RHB HB FB TB LWB LSB)`. This list is a preference ordering. `obj3` will most often be the RFL, then the RWB, and so on. Given the preference orders for all objects, a *consistent* assignment of trajectory data to the play description must be made. Here our system finds the single most consistent interpretation using preference assignments, the constraint that all trajectories must be assigned to an object in the temporal structure description, and a heuristic scoring function. Due to space limitations this matching process is not discussed further.

3.4 Temporal constraints

The remaining slots in the the temporal structure description indicate the temporal and logical relationships between agent goals. Two temporal primitives are available: *before* and *around*. For example, "(before obj1_act1 obj1_act2)" indicates that goal obj1_act1

[2] For example, passPatCutting takes parameters (obj a toReg inReg). The network encodes detects the following: *Obj*, which must be an eligible receiver, runs a pass pattern segment making a sharp (e.g. about *a* degrees) change in motion in *inReg* after which *obj* is moving in towards the *toReg*.

occurs before obj1_act2, where obj1_act1 is the label for "snapToQB (obj1)" and obj2_act2 is the label for "blockQBPass (obj1)". Similarly, "(around obj3_act2 obj4_act2)" indicates that object3's passPatCutting goal occurs around the same time as object4's passPatCutting goal. The meanings of "before" and "around" will be defined shortly. Finally, "(xor obj3_act5 obj4_act4)" indicates that object3's catchPass goal xor object4's catchPass goal should be observed.

By assumption, the goals of an agent are active during temporal intervals of finite duration; they are not instantaneous events. As such, Allen's interval algebra [1] applies and there are potentially 7 possible temporal relations (not counting inverses). However, that algebra requires precise definition of the endpoints of the intervals. Our ability to assign goals to agents based upon perceptual evidence will be fuzzy, allowing us only to assign a graded value that varies over time. In the ideal case there would be a nice peak or plateau in the probability a goal is active during a temporal window, but real data is rarely ideal.

Note that our temporal constraints do not support most temporal implications. For example, the temporal relation of simultaneity is expressed as `around` which can be interpreted as "about the same time as." Clearly such a 'fuzzy' relation is not transitive and we cannot apply transitive closure to the temporal relations. Rather, we only exploit those relations manually constructed by the knowledge engineer designing the action description.

4 Visual nets and temporal functions

Previous work has shown that agent goals can be represented in a probabilistic framework using Bayesian belief networks [4, 11, 16]. We also use belief networks based on visual evidence, or *visual networks*, that offer a rich representation designed to handle uncertainty in evidence, goal models, spatial reasoning, and temporal reasoning. Further, the networks can be used as building blocks for recognizing multi-agent activity.

4.1 Network structure and evaluation

A single belief network represents each goal or event and can be instantiated at any time during a play. The networks typically contain between 15 and 25 nodes with a relatively tree-like link complexity and therefore exact propagation algorithms can be used to compute the probabilities of each node state [15]. The structure of each network is manually specified. Currently the priors are also manually assigned, however some priors can be obtained from analyzing the evidence and the performance of particular feature detectors.

Figure 4 shows one such network, `catchPass`. The network consists of two types of nodes: unobservable belief and observable evidence.

Unobservable belief nodes A belief node has two states, *true* and *false*, and represents an internal state of the agent or some external state in the world at the time when the network is evaluated. Each visual network has a designated *main goal node* (e.g. `catchPass`).

Observable evidence nodes An evidence node's states and state values are directly dependent upon the data. Some nodes are binary (e.g. *observed*, *notObserved*), most are trinary, (e.g. *observed*, *maybeObserved*, *notObserved*), and the remainder have specialized states that quantize a particular feature detector output (e.g. the result of the `distance` detector is quantized into states *inContact*, *nextTo*, *near*, *inVicinity*, *far*, *distant*). To maintain continuous valued information, whenever possible evidence is entered as "virtual" likelihood evidence.[3]

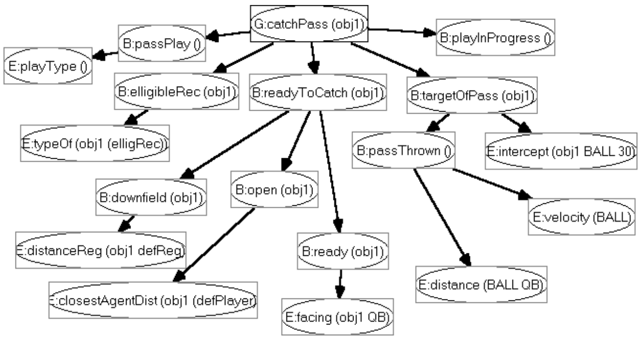

Figure 4: The `catchPass` goal network.

The main belief node of each network can accept parameters set by the caller of the network at run-time. For example, goal node `catchPass (obj1)` accepts one argument, a specific agent. Each network is designed so that it can be applied to any world object and return a reasonable result.

4.2 Locality in space-time

Visual networks can be applied to any agent at any time. As much as possible, visual goal networks are designed to use evidence observed locally in space and time. Further, evidence features are typically deictic, or agent centered. For example, networks sometimes compute the distance between the current agent and the *closest agent*.

Because goal networks can make use of dynamic state variables (e.g. `snapTime`) and the output of other goal networks (e.g. `catchPass` uses the result of the `playInProgress` network), the networks are not entirely "closed." Incorporating input from other networks or dynamic state variables violates the belief network assumption that all variable dependencies are modeled via explicit conditional probabilities. We accept this approximation, noting that the networks themselves are simplified approximations to the actual dependency structure and that partitioning actions into small networks simplifies and makes manageable the job of the knowledge engineer.

We incorporate evidence from an external network, such as the `playInProgress` evidence node, into a network such as `catchPass (obj1)` as follows. If the `playInProgress` network cannot evaluate and returns NIL, no evidence is entered for the node. If the `playInProgress` network returns a high likelihood of a particular state that exceeds a predetermined threshold for `playInProgress`, evidence is entered directly into the `catchPass` network (e.g. if *observed* = .99 and *notObserved* = .01 and `threshold(playInProgress) = .85` then *observed* = 1.0 is entered into `catchPass`). Finally, if `playInProgress` evaluates below the threshold, the beliefs are treated as direct evidence and the probabilities are converted

[3] So-called "virtual" evidence, or the relative likelihood of each of the discrete states, is entered into a network to use continuous-valued evidence in a node with discrete evidence states (see [15]). The likelihood is obtained using the relative activation levels of each discrete state which are computed with piecewise linear functions.

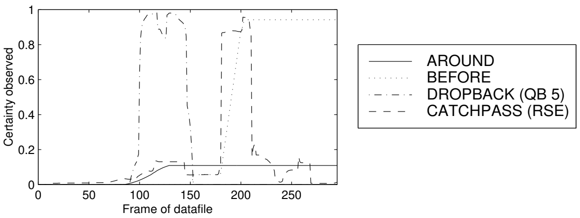

Figure 5: Goal likelihood curves returned by the networks "dropback (QB 5)" and "catchPass (RSE)" superimposed with the corresponding temporal curves for "dropback (QB 5) before catchPass (RSE)" and "dropback (QB 5) around catchPass (RSE)".

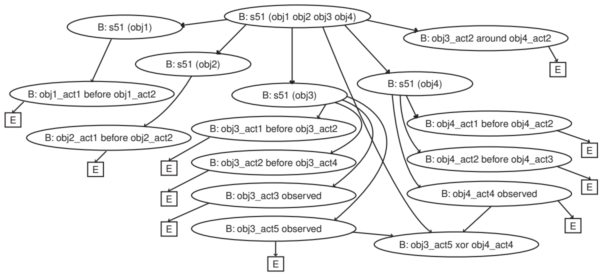

Figure 6: The s51 multi-agent recognition network.

to likelihood evidence [15] (e.g. if *observed* = .8 and *notObserved* = .2 and `threshold(playInProgress)` = .85 then the evidence that *observed* is 4 times more likely than *notObserved* will be entered into the `catchPass` network).

4.3 Temporal analysis functions

The output of a visual goal network at each frame for a given object results in a likelihood curve over time. Temporal relationship evidence detectors use these curves as input. The functions compute a certainty value for the *observed*, *before*, and *around* tests at each time frame using heuristic functions that compare the activation levels of each goal over time, characteristics of each input curve, the temporal distance between features of the curves, the amount of overlap between the curves, and a minimal activation time for each goal. The functions are designed to preserve the uncertainty in the output of the visual goal networks and to avoid hard thresholding. Two curves returned by the networks "dropback (QB 5)" and "catchPass (RSE)" are shown in Figure 5 overlaid with the likelihood values for the *before* and *around* detectors corresponding to "dropback (QB 5) before catchPass (RSE)" and "dropback (QB 5) around catchPass (RSE)".

5 Multi-agent networks

Multi-agent action is recognized using a multi-agent belief network. At each time, the network integrates the likelihood values returned by temporal analysis functions at that time and returns a likelihood that a given play has been observed.

Figure 6 shows an example of a multi-agent network for the s51 play. The network structure is generated *automatically* from the temporal structure description. In the system discussed in this paper, a two-level naive Bayesian classifier network structure is generated that encodes the temporal structure of a play. All nodes in the multi-agent networks represent beliefs or evidence observed over all the play data seen from the start of the play until the current time. The state characterization of all nodes comprises the values $(observed, notObserved)$. The main node in the example is *B: s51 (obj1 obj2 obj3 obj4)*. Linked to that node is one node for each agent – for example *B: s51 (obj1)* – representing the belief that the agent's goals for the s51 have been observed. Below these nodes are nodes representing:

- Binary temporal relationships between goals (e.g. *B: obj1_act1 before obj1_act2*). These nodes represent the belief that a particular temporal ordering has been *observed* or *notObserved* at some point during the action sequence.
- Evidence for binary temporal relationships (e.g *E: obj1_act1 before obj1_act2*). There is a conditional link from the temporal relation belief node to the evidence. The evidence values are computed by the temporal analysis functions. To avoid cluttering the figure, these nodes are represented with a boxed "E" node.

Temporal relationships between agents are linked directly to the top-level belief node (e.g. see *B: obj3_act2 around obj4_act2*). Additional links can be added for logical relationships, which conditionally link the two related goal observations.

A detector such as *E:obj3_act1 before obj3_act2* implicitly encodes the observation *E:obj3_act1 observed* and *E:obj3_act2*. Therefore, when an agent goal node is temporally compared to some other agent goal node, only the temporal comparison belief node is incorporated into the network. However, some goal actions are not included in any temporal comparisons in the temporal action description. In these cases, the network includes an *observed* belief and evidence node (e.g. *B:obj3_act3 observed*).

Conditional and prior probabilities for the network are determined automatically using heuristics matching table templates to specific node-link combinations, similar to the method used by Huber [11]. The structure of the network for the s51 shown in Figure 6 essentially implements a weighted voting scheme between observed goals and temporal relationships between goals.

Experimental evaluation has demonstrated that naive Bayesian networks are surprisingly good classifiers, despite making strict independence assumptions between attributes and the class. Moreover, recent work has shown that augmenting such networks with additional binary conditional dependencies improves classification performance so that it is often better and otherwise comparable to more complex representations, including more highly-connected learned network structures [7]. Our multi-agent networks are naive classifiers where binary temporal relations between goals have been encoded *within* nodes, not in links between nodes.

The network shown in Figure 6 is only for a play with four agents where the number of actions for each agent is restricted to just a few examples. For a play with 11 agents, the networks typically contain *at least* 50 belief nodes and 40 evidence nodes and often twice that number. Network propagation by exact algorithms is feasible, however, because the network has a shallow tree linking structure and consists of binary internal belief nodes. The temporal analysis functions return continuous valued likelihood information. This information is entered into the multi-play network as continuous evidence, avoiding unnecessary thresholding of uncertain information.

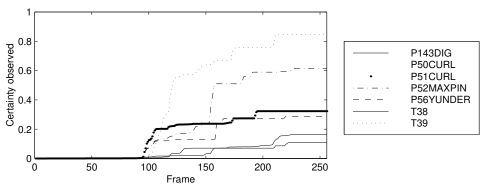

Figure 7: Result of running 7 play detectors on a t39 play example. Shown is the likelihood of each play having been observed at frame t considering all evidence from frames $0 - t$.

6 Results

We are using the representation described in this paper in a football play recognition system. The system has knowledge of about 40 region definitions (e.g. `line-of-scrimmage`), 60 player types (e.g. `quarterback`, `receiver`), and `ISA` relationships between player types (`wide-receiver ISA receiver`). We have constructed approximately 60 evidence detectors (e.g. `distance(closestAgent)`) that are applied to the trajectory data and produce probabilistic quantized outputs (e.g. *inContact* = 0.3, *nextTo* = 0.7). We estimate 70 robust visual networks will ultimately be required for recognition of most of the plays in our database, and about 50 of those have been constructed.

We have evaluated our system on 29 tracked plays using a database of 10 temporal play descriptions. Figure 7 shows the likelihood value obtained by evaluating the multi-agent network at each frame for 7 play models on a datafile for a t39 play. Here the desired behavior is achieved: uncertain evidence of temporal relationships between goals is sufficient to cause the t39 play detector's likelihood value to quickly rise above the other plays shortly after the play action begins at frame 90.[4]

Figure 8 is a confusion matrix showing the final likelihood value obtained for each temporal play description when run on 29 example plays. A "-" value indicates a play where no good object-to-trajectory consistency match could be found.[5] The examples below the line (i.e. p58 through s35) do not yet have fully implemented temporal play descriptions. The highest likelihood value obtained on each data file (each row) is marked in bold.

Considering only the top portion of the table, the maximum likelihood value along each row selects the correct play for 21 of the 25 play instances. 3 of the 4 errors are caused by p56yunder examples being misclassified as p52maxpin plays. Figure 9, which shows the diagrams for those two plays with a misclassified example approximately overlaid on top demonstrates why the system has difficulty classifying the example. The diagram shows that both plays, when executed perfectly, are similar when the "optional action" is not taken into account. The only large observed difference between the plays is for the rightmost player, who follows a trajectory different from both the p56yunder and the p52maxpin. Our models currently do not include the optional actions, which would contribute evidence to the desired p56yunder classification. We are currently extending the multi-agent networks so they can

[4] The system requires approximately 1 second of computation per frame per tested play on a 500 MHz Digital Alphastation and could be highly parallelized.

[5] Prior to evaluating a particular multi-agent network, a consistent match between the labeled trajectories and the object label preference orderings must be found. This component of the system is not discussed in this paper.

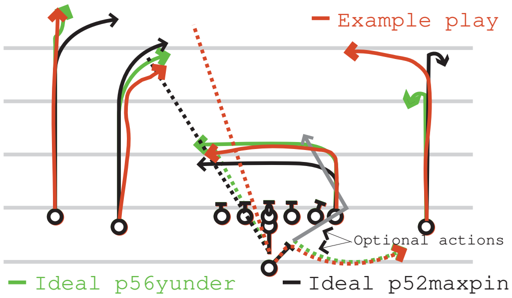

Figure 9: P56yunder and p52maxpin play diagrams with one p56under example play approximately overlaid. The system returned likelihoods of .64 for p56yunder and .76 for p52maxin.

encode optional compound goals.

The bottom section of the table are the probabilities produced when applying the system to instances of plays for which there is (as yet) no action network. The discouraging result here is that false positives have values comparable to the correct positives above. That is, while our current system is capable of selecting the correct play description, it cannot yet determine when a play does not belong to one of its known categories. One reason for this is that we have not yet completed constructing all the visual networks necessary to provide rich descriptions of the plays. The weaker the model, the more easily it is matched by some incorrect instance. More detailed models will improve the ability of the system to determine that a play is "none of the above."

Overall the results are promising, especially considering the complexity and variation of the input data. We have data to evaluate additional play descriptions but must first complete coding the additional goal networks. Further, the multi-agent belief networks need to be extended to handle compound groups of actions (e.g. player performs `(XOR (goal-a and goal-b) (goal-c and goal-d))` before we can completely characterize the competence of the representation.

7 Final remarks

We have proposed a representation – motivated by findings in the computer vision object recognition literature and the power of augmented naive Bayesian classifiers – that represents complex, multi-agent action using low-order temporal graphs. The primitives in these graphs are agent-based belief networks that can recognize agent goals by probabilistic integration of visual evidence. Networks with a structure similar to naive classifiers are automatically generated from a simple description of a team play. These networks contain nodes that encode binary spatial and temporal relationships and are small and therefore computationally manageable. We have demonstrated that these networks can recognize multi-agent action for a real domain with noisy input trajectory data. Studying the representational, recognition, and computational properties of the multi-agent networks is the focus of our current work.

References

[1] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

Name	p143dig	p50curl	p51curl	p52maxpin	p54maxcross	p56yunder	p63up	p63upa	t38	t39
p143dig (file aa00185)	**.75**	.49	-	-	.37	.33	-	.24	.53	-
p143dig (file aa00412)	**.98**	.63	-	-	.75	.71	-	.57	.65	-
p143dig (file aa00606)	**.93**	.45	-	-	.57	.63	-	.32	.39	-
p143dig (file aa00847)	**.87**	.35	-	-	.53	.49	-	.27	.30	-
p143dig (file aa01032)	**.91**	.42	-	-	.50	.36	-	.60	.41	-
p143dig (file aa02128)	**.86**	.42	-	-	.43	.41	-	.70	.43	-
p143dig (file aa02329)	**.98**	.58	-	-	.85	.65	-	.57	.36	-
p50curl (file aa06046)	.19	**.87**	-	-	-	.44	-	.62	.58	.27
p51curl (file aa10542)	-	.21	**.69**	-	-	.27	.35	.34	-	.58
p51curl (file aa10736)	-	.54	**.95**	-	-	-	-	.55	-	.66
p51curl (file aa11033)	-	-	**.98**	-	-	-	-	.82	.09	.68
p52maxpin (file aa14122)	-	-	.37	**.93**	-	.66	.88	-	-	-
p54maxcross (file aa19487)	.55	.55	.37	.57	**.97**	.48	-	.77	-	-
p56yunder (file aa28294)	-	-	.47	-	-	**.63**	-	-	-	-
p56yunder (file aa29325)	-	-	.24	.51	-	**.69**	.39	-	-	-
p56yunder (file aa29486)	-	-	.75	**.88**	-	.83	.72	-	-	-
p56yunder (file aa30045)	.61	.26	-	-	-	**.80**	-	.73	.41	.47
p56yunder (file aa30560)	.38	-	.38	**.78**	-	.62	.57	-	-	-
p56yunder (file aa30761)	-	-	.54	**.76**	-	.64	.34	-	-	-
p63up (file ab00958)	-	.41	.56	-	-	-	**.87**	-	-	-
p63up (file ab01196)	-	.61	.79	-	-	-	**.95**	-	-	-
p63up (file ab01570)	-	.35	.43	-	-	-	**.89**	-	-	-
p63upa (file ab00636)	-	-	.52	-	-	-	-	.73	.12	**.76**
t38 (file bb23079)	-	-	-	-	-	-	-	.27	**.83**	.51
t39 (file bb31597)	-	.25	.39	-	-	.27	.55	.30	-	**.83**
p58 (file aa36188)	-	-	-	.30	-	-	**.57**	-	-	-
r34wham (file ba28768)	.35	.62	-	-	.42	.43	-	**.65**	.56	-
s14wham (file ba45295)	-	-	-	-	-	.27	-	.57	**.72**	.53
s35 (file bb05291)	.16	.45	.22	**.64**	-	.31	.40	-	-	-

Figure 8: Likelihood values when the recognition system is run for each of 10 play models over a dataset of 29 examples.

[2] J. Azarewicz, G. Fala, and C. Heithecker. Template-based multi-agent plan recognition for tactical situation assessment. In *Proc. of the Sixth Conference on Artificial Intelligence Applications*, pages 248–254, 1989.

[3] H. Buxton and S. Gong. Advanced visual surveillance using Bayesian networks. In *Proc. of the Workshop on Context-Based Vision*, pages 111–123, Cambridge, MA, June 1995. IEEE Computer Society Press.

[4] E. Charniak and R. P. Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, 64:53–79, 1993.

[5] M. Devaney and A. Ram. Needles in a haystack: Plan recognition in large spatial domains involving multiple agents. In *Proc. Fifteenth Nat. Conf. on Artificial Intelligence*, pages 942–947, Madison, WI, July 1998.

[6] J. Forbes, T. Huang, K. Kanazawa, and S. Russell. The BATmobile: towards a Bayesian automated taxi. In *Int'l Joint Conf. on Artificial Intelligence*, volume 14, pages 1878–1885, 1995.

[7] N. Friedman and M. Goldszmidt. Building classifers using Bayesian networks. In *Proc. Nat. Conf. on Artificial Intelligence*, pages 1277–1284. AAAI Press, 1996.

[8] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[9] B.J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[10] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proc. Nat. Conf. on Artificial Intelligence*, pages 966–972. AAAI Press, 1994.

[11] M.J. Huber. *Plan-Based Plan Recognition Models for the Effective Coordination of Agents Through Observation*. PhD thesis, University of Michigan, 1996.

[12] S.S. Intille and A.F. Bobick. Closed-world tracking. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 672–678, Los Alamitos, June 1995.

[13] H.A. Kautz and J.F. Allen. Generalized plan recognition. In *Proc. Nat. Conf. on Artificial Intelligence*, pages 32–37, August 1986.

[14] H.-H. Nagel, H. Kollnig, M. Haag, and H. Damm. Association of situation graphs with temporal variations in image sequences. In *Computational Models for Integrating Language and Vision*, pages 1–8, November 1995.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[16] D.V. Pynadath and M.P. Wellman. Accounting for context in plan recognition, with application to traffic monitoring. In P. Besnard and S. Hanks, editors, *Int'l Conference on Uncertainty in Artificial Intelligence*, volume 11, 1995.

[17] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *Proc. Int'l Conf. Computer Vision*, volume 6. IEEE Computer Society, Narosa Publishing House, January 1998.

[18] G. Retz-Schmidt. A REPLAI of SOCCER: recognizing intentions in the domain of soccer games. In *Proc. European Conf. AI*, volume 8, pages 455–457, 1988.

[19] M. Tambe. Tracking dynamic team activity. In *Proc. Nat. Conf. on Artificial Intelligence*, pages 80–87, August 1996.