M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 454
January, 1998

Representation and Visual Recognition of Complex, Multi-agent Actions using Belief Networks

Stephen Intille **Aaron 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 logical and 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.
Keywords: vision, knowledge representation, plan recognition

1 Introduction

As computer vision begins to address increasingly dynamic domains, for instance video annotation and automatic surveillance, the general task of recognition grows to encompass not only understanding simple motion but also identifying large scale actions [22, 4]. As it does, the methods of recognition can no longer be supported solely by measurement and geometry or by precise CAD-like models, as is typical of static object recognition. 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, 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). Later we will precisely describe the temporal and logical relations we will consider, as well as what can constitute an action component `A`. For now we note that the relations generally reflect causal connections or influences between components. An additional source of complexity is that 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 three aspects of the problem. The first is that the temporal constraints of the action description itself are typically probabilistic: e.g. `B` follows `A`, but only 80% of the time. This uncertainty reflects that complex actions are not typically defined by necessary or sufficient conditions but rather are comprised of typical components. The second source of uncertainty is the fuzziness of attributes that must be assigned to individual elements, e.g. `obj1` is `near obj2`. Finally, as the design of the representation is intended to support recognition, we 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, animal group behavior (e.g. video of large public gathering), interactive environments, 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 [16]. 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 manually tracked plays.[1]

1.1 An analogy to complex 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 [9]. 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

[1] The trajectories are obtained by following players and field points with a mouse in panning and zooming video of a play. The process of warping the noisy data using the tracked field coordinates to the field coordinate system introduces a substantial amount of noise into the trajectory data.

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 [10] 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 [10] 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.

1.2 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 and logical constraints to be verified.
- For each basic element of the temporal structure, we define a *visual network* that probabilistically detects the occurrence of the individual goal or event at a given time. These networks encapsulate the knowledge required to address the individual decisions; the networks may refer to other such networks, but the more they do the less they decouple the individual goals. The evidence nodes of these graphs are direct perceptual sensors, such as the quantized relative position of two players.
- Temporal analysis functions are defined which evaluate the validity of a particular temporal relationship. For example, if there is a visual network detecting a quarterback throwing the ball and another detecting a receiver catching it, the temporal analysis functions can verify that one local event happened `before` the other.
- A large multi-agent temporal goal belief network is automatically constructed reflecting the temporal and logical constraints of the temporal structure. The nodes are the individual goals and the specified temporal and logical relations; the links enforce conditional probabilities (and associated logical relations such as `xor`).

Multi-agent action recognition is accomplished by evaluating the appropriate belief networks.

The remainder of the paper describes the representational components in more detail, in particular identifying the elements of uncertainty each is designed to accommodate.

2 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 and logical constraints. In designing such a structure we are explicitly stating that the complex actions we are representing have such a prototype, and that it can be expressed with this language. Furthermore, since the description needs to be generated manually, the description needs to be simple to express. We will show that this is the case for highly-structured yet complex actions such as football plays. Which additional domains are amenable to such description is currently a focus of our research.

2.1 Individual goals and behaviors

Intuitively we need to express constraints such as "The quarterback hands off the ball *before* the halfback runs behind the left tackle." To do so we need to define the individual elements between which we establish the temporal and logical relations. What are the equivalent of "model edges" in object recognition?

Within our current framework, we use individual agent *goals* as the basis for the descriptive structure. We view complex actions as a partially ordered set of goal directed behaviors on the part of interacting agents. However, unlike [11] and other work where logics of intention and communication are proposed to determine if an agent's behavior is logically consistent with a given goal, we will *define* goals by their (probabilistic) characteristic behaviors, building on work in probabilistic plan recognition [6]. 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. Later we will more fully detail the construction of belief networks that serve as the definition of the individual agent goals.

A *compound goal* is simply a goal comprised of multiple primitive goals. Temporal and logical relations can exist between the compound goal and other primitive goals, as well as between the primitive goals of the compound goal itself.

Since compound goals could be expanded by their definition in terms of their primitive goals, one might wonder why use them at all. The main reason is that compound goals represent causally linked individual goals that are tightly coupled and often found in other temporal structure descriptions. For example, the compound goal "catchPassAndRunWB" is, unsurprisingly, common to many pass plays. Having a single compound entity facilitates generating the temporal descriptions. They also facilitate the assignment of conditional probabilities, since data from across many plays can be used to construct them.

```
(teamGoal p51curl
        (name "p51curl")
        (agentGoal ifl
            (name "p51curl-FL")
            (goal ifl1 "setInPosition (FL FLRegion)")
            (goal ifl2 "zoomMotion (FL)")
            (goal ifl3 "cutDownfield (FL)")
            (goal ifl4 "runCirclePat (FL 5)")
            (goal ifl5 "runInFront (FL WB FL)")
            (goal ifl6 "turnToward (FL QB)")
            (goal ifl7 "lookForPass (FL)")
            (goal ifl8 "catchPass (FL)")
            (goal ifl9 "blockForBc (FL)")
            (before ifl1 ifl2 ifl3 ifl4 ifl5 ifl6 ifl7)
            (xor ifl8 ifl9)
            (before ifl7 ifl8)
            (before ifl7 ifl9))
        (agentGoal iwb
            (name "p51curl-WB")
            (goal iwb1 "setInPosition (WB WBRegion)")
            (goal iwb2 "runOutPat (WB 5)")
            (goal iwb3 "runBehind (WB FL)")
            (goal iwb4 "lookForPass (WB)")
            (compoundGoal iwb5
                (name "catchPassAndRunWB")
                (goal iwb6 "catchPass (WB)")
                (goal iwb7 "progressDownfield (WB)")
                (before iwb6 iwb7))
            (goal iwb8 "blockForBc (WB)")
            (before iwb1 iwb2 iwb3 iwb4)
            (xor iwb5 iwb8)
            (before iwb4 iwb5)
            (before iwb4 iwb8))
        (agentGoal iqb
            (name "p51curl-QB")
            (goal iqb1 "setInPositon (QB QBRegion)")
            (goal iqb2 "recieveBallSnap (QB)")
            (goal iqb3 "dropback (QB 5)")
            (goal iqb4 "findOpenRec (QB right)")
            (goal iqb5 "throwPass (QB)")
            (before iqb1 iqb2 iqb3 iqb4 iqb5))
        (around ifl1 iwb1 iqb1)
        (around iqb2 ifl4)
        (before iqb5 iwb6)
        (before iqb5 ifl8)))
```

Figure 1: A simplified (i.e. only some agents and actions) temporal structure description for the `p51curl` play.

2.2 Temporal and logical constraints

We describe the temporal structure description using an example. Figure 1 gives part of the temporal structure description for the play `p51curl`; only some goals for 3 of the 11 offensive players are listed. Note that the description includes definitions of compound goals, such as `iwb5`, `catchPassAndRunWB`.

2.2.1 Temporal relations

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 [2] 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.

Furthermore, fuzzy temporal intervals prevent temporal implications. For example, in our system 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 can not support hard chaining. While perhaps we could implement some form of fuzzy transitivity, such a method would not overcome the difficulty of propagating probability. For example, suppose for a given play we know the probabilities of `A before B` and `B before C`, where `A`, `B`, and`C` are individual agent goals. How should these be combined to determine the probability of `A before C`? Clearly they are unlikely to be independent given that both involve agent goal `B`.

Because of this imprecision, we do *not* apply transitive closure to the temporal relations. Rather, we only exploit those relations manually constructed by the knowledge engineer designing the action description. The only temporal relations in the temporal structure graph are the `before` and `around` relations explicitly coded.

The syntax of `around` and `before` is that they take an arbitrary number of goals; the resulting relations are taken between neighbors in the list. `(around A B C)` implies `(around A B)` and `(around B C)`, but not `(around A C)`.

2.2.2 Logical relations

The remaining components of the description are the logical elements operating on goals, of which there are only two: `xor` and `and`. The semantics are standard: `xor` requires that exactly one element of a set of goals be achieved; `and` requires all of them. These logical relations will be converted into biases on conditional probabilities when we address the issue of uncertainty. Furthermore, in the absence of logical relations, component goals of compound goals are considered conditionally independent providing independent evidence in support of their predecessor compound goal.

3 Visual networks

Previous work has shown that agent goals can be represented in a probabilistic framework using Bayesian belief networks [6, 15, 25]. 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.

3.1 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 [17]. 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 2 shows two such networks, `catchPass` and `playInProgress`. The networks consist 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. Belief nodes can also represent the belief that an agent has a particular goal; these nodes are not attached directly to evidence

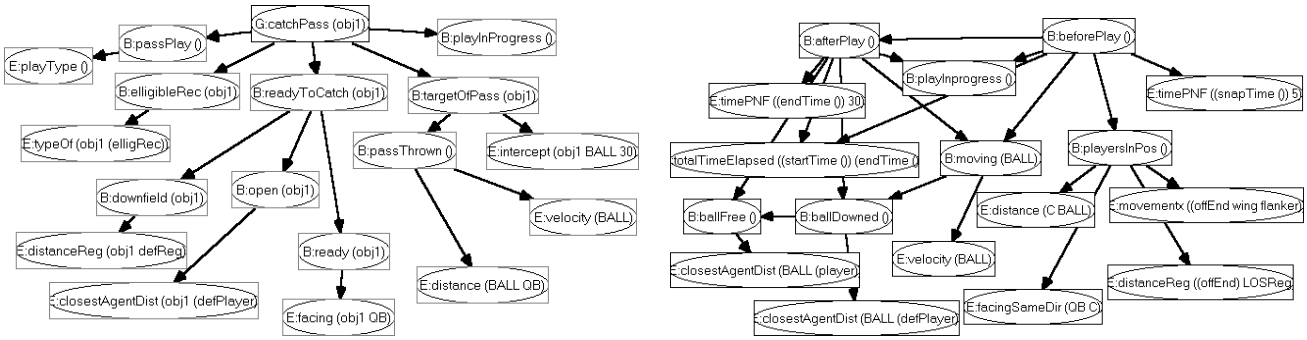

Figure 2: Two visual networks: the `catchPass` goal network (left) and the `playInProgress` belief network.

nodes. This distinction forces the network designer to explicitly represent two types of uncertainty: (1) the uncertainty of the detectors given noisy data, modeled in the conditional probabilities $P(evidence|belief)$, and (2) the uncertainty resulting from the use of an approximate model, which is modeled in the conditional probabilities $P(belief|beliefInGoal)$. This conceptual distinction can simplify the knowledge engineering. Each visual network has a designated *main belief node* (e.g. `catchPass` and `playInProgress`).

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 quantizes a particular feature detector output (e.g. the result of the `distance` detector is quantized into states *inContact*, *nextTo*, *near*, *inVacinity*, *far*, *distant*). Each evidence node requires a specialized function. When the data is inconclusive, the function can return a *NIL* value, and the evidence node will not be instantiated with any value in the network.

Evidence nodes can take objects, groups of objects, object categories, regions, or times. For instance, the `distanceReg ((offEnd) LOSReg)` node in the `playInProgress` network will compute the distance from any offensive end in the play (of which there are several possible types) in the scene with the line-of-scrimmage (LOS) region. Most evidence functions will work with groups of agents. `velocity (LT RT LG)` will return the average velocity of the three objects. Some regions and functions (e.g. `LOSReg` and `snapTime()`) are dependent upon variables that have been previously set by other networks.

By nesting functions, fairly complex references to surrounding agents can be made from within a network. For example, to compute the distance from the agent to the closest defensive agent, the detector `distance (closestAgent (defense))` is applied.

All references to other agents are relative to the position of the current agent (e.g. closest agent, second closest agent, agents within some distance, etc.)

The main 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.

3.2 Approximations to the closed-world principle

Since 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." External knowledge can impact the network, which violates the Bayesian assumption, leading to inexplicit dependencies. We accept this approximation, noting that the networks themselves are an approximation to begin with and that partitioning actions into small networks simplifies and makes manageable the job of the knowledge engineer. Later in this paper we discuss some control issues raised by network interdependence.

Evidence from a second network, such as the `B:playInProgress` evidence node in the `G:catchPass (obj1)` network, is incorporated 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 probability 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 resulting probabilities are converted to "virtual" likelihood evidence [24] and entered into the network (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 Multi-agent networks

Previous work on representing collaborative activity can be divided into methods that "compile" down team goals or intentions and those that have an explicit representation for group intentionality [11, 7]. Theories of group intentionality have proven difficult to implement in practice and still lack practical algorithms for governing not only the need for communication, but also the *utility* of

communication [32]. Further, work on modeling group intentionality does not generally address the *visual recognition* of group intention, where agents are acquiring much of their information about the world through probabilistic, perceptually-based action recognition. Our hypothesis is that visual networks and the temporal structure description can, when integrated probabilistically, be used to recognize structured but complex team goals in some interesting multi-agent domains without explicit reasoning about group intentionality and communication. This integration is accomplished by a special extension of visual networks to *multi-agent networks*.

4.1 Multi-agent network structure

Figure 3 shows an example of a multi-agent network for recognizing part of the action description shown in Figure 1 for the `p51curl` play. The network structure is generated automatically from the temporal structure description. It is similar to the visual networks described previously, but it has some special structure, described below, to facilitate recognition of group action by multiple agents.

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. All nodes have the state $(observed, notObserved)$. The main node in the example is `Belief p51curl`. Linked to that node is one node for each agent (e.g. `Belief p51curl-QB`) representing the belief that the agent's compound goals for the p51curl have been observed. Below these nodes are nodes representing:

- Belief nodes for the agent's prototypical goals for the play (e.g. `Belief receiveBallSnap (QB)`). These nodes encode the belief that goals will be *observed* or *notObserved* at some point during the given play and are therefore each conditionally dependent on their predecessor compound goal belief (e.g. `Belief receiveBallSnap (QB)` links from `Belief p51curl-QB`) but conditionally independent of each other unless explicitly logically connected.
- Evidence that particular goals have been active during the observed sequence (e.g. `Evidence observed receiveBallSnap (QB)`). Each such evidence node has a corresponding visual goal network which, which evaluated at a particular time, produces a probability that goal is active at that time. By heuristically checking for peaks in the observed visual goal network output over the observed time period, evidence that a goal has been *observed* or *notObserved* can be recovered. By tuning the conditional probability table, $P(evidence|belief)$, sensor uncertainty is modeled.
- Logical relationships between belief in multiple goals and/or compound goals (e.g. `XOR` between `Belief catchPassAndRunWB` and `Belief blockForBC (WB)`). The two logical relations, `XOR` and `AND` are represented using nodes and the appropriate conditional probability tables that enforce logical consistency. These nodes are conditionally dependent on their predecessor compound node.
- Binary temporal relationships between goals (e.g. `Belief (dropback (QB 5)) before (throwPass (QB))`) Belief that a particular temporal sequence has been observed during the action sequence, which is conditionally-dependent on the predecessor compound goal and each of the specified goal actions.
- Evidence for binary temporal relationships (e.g `Evidence (dropback(QB 5)) before (throwPass (QB))`). As with the goal evidence observed nodes, there is a conditional link from the temporal relation belief node to the evidence. The implementation of the fuzzy temporal evidence functions will be discussed below.
- Subgoals that are themselves compound goals (e.g. `Belief catchPassAndRunWB` and `Belief p51curl-QB`). A compound node links to each of its subgoals and links from its predecessor compound goal. The semantics of compound goal temporal relationships (e.g `Belief observed catchPassAndRunWB` before `Belief observed blockforBC (WB)`) are discussed shortly.

There is one exception to the rule that a goal belief node is conditionally-dependent on its predecessor compound goal belief node. Each agent's play goal node (e.g. `Belief p51curl-QB`) links *to* the main play goal node. We assume that in a multi-agent world, situations that affect the ability of the team to achieve the goal will most often affect local components of the goal, specifically *particular agents*. The conditional probability table $P(p51curl|p51curl\text{-}WB, ..., p51curl\text{-}QB)$, therefore, can be used to implement a logical gating function that represents knowledge like "9 out of 11 agents should be achieving their goals, but the `QB` and `FL` *must* be achieving their goals for a `p51curl` to exist."[2]

Over time, the chance that agents will have problems achieving their goals will increase. We can model this plan deterioration by appropriately adjusting the conditionals, $P(action|agentPlaySubgoal)$, to model the uncertainty in temporally distant action in the temporal structure prototype description.

Not shown in Figure 3 are additional play context, or precondition, nodes. For instance, for the `p51curl` to be active, no interception should have been observed. Beliefs and evidence that check for this conditional, just like those in visual networks, are hung *below* the play goal node as needed.

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[14]. Our current research focuses on tuning and evaluating and the resulting multi-agent networks.

The network shown in Figure 3 is for two players, where some actions have been removed for clarity. For a play with 11 agents, the networks are quite large, although propagation by exact algorithms remains feasible because the networks are relatively shallow, sparsely connected, and consist of binary nodes.

4.1.1 Implementing fuzzy temporal relationships

Temporal relationship evidence detectors use heuristic functions that analyze the probabilistic output (over a

[2] To reduce table size, special heuristics analogous to the "noisy-or" are being used.

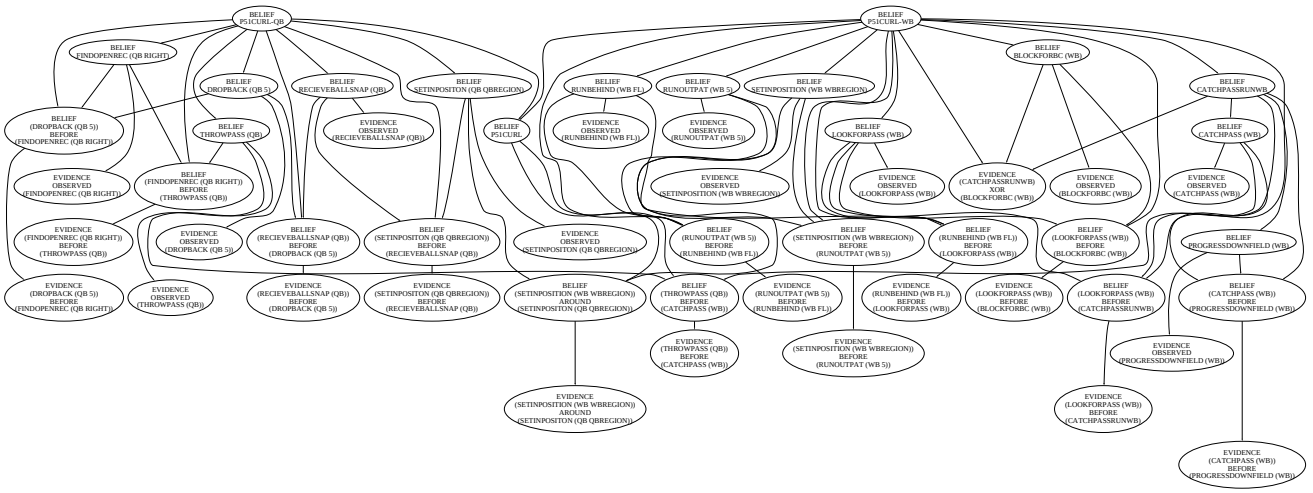

Figure 3: A multi-agent network structure for part of the description in Figure 1. The arrows always point downwards.

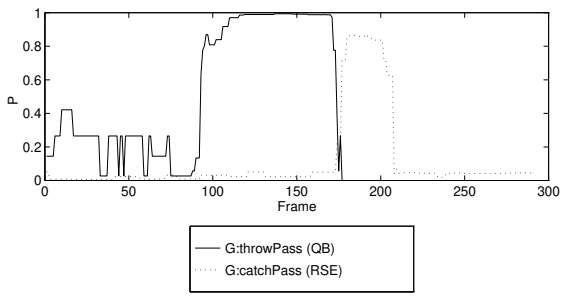

Figure 4: The output of the `throwpass (QB)` and `catchpass (RSE)` visual goal networks.

window of time) of the goals being compared. Based on characteristics of each output curve, the temporal distance between features of the curves, and a threshold value for each goal, the functions compute the evidence for the `before` and `around` conditions. We are currently working to extend the heuristic functions to output probabilistic (as opposed to quantized) values and to use more robust methods of comparing the goal probability curves. Figure 4 shows the output of the `throwpass (QB)` and `catchpass (RSE)` visual goal networks at each time for a particular example of the `p51curl` play. The temporal detector `evidence (throwPass (QB)) before (catchPass (RSE))` will return *observed* given such data.

We interpret the semantics of `before A B`, where B is a compound goal, to mean that A is `before` *all* the subgoals of `B`. Similarly, for around, `A` is `around` all the subgoals of `B`. The activation evidence from each of the subgoals of `B` at each time is compared and the maximum value is returned. The resulting curve (the maximum activation of any subgoal at each time) is then use to compute temporal relations using the heuristic functions.

4.2 Recognition and control

Missing from our framework as described so far is a method of controlling the amount of computation that must be performed at each time point and actually using the multi-agent networks. There are three issues: iterative computation, expectation, and comparison.

Both the single and multi-agent networks can contain nodes that refer to other networks or state variables that are set by other networks. While the knowledge engineer is encouraged to minimize the number of interconnected networks, the system needs a strategy for dealing with those that remain. At each time step, there are three options: (1) compute all the networks and then iterate, hoping for convergence, (2) compute all the networks, recomputing networks whenever they are encountered as sub-nodes, avoiding loops, and (3) compute only the networks that are needed, caching any computed values to be used in other networks, possibly ordering computation using offline sensitivity or network interconnectivity analysis. Currently we use the second strategy.

To use the more desirable third strategy, the system needs to know which goals to expect given a particular situation. We are investigating using the dependencies encoded in the temporal structure description combined with the output of the multi-play networks to limit computation of goals and perhaps the size of the multi-agent networks that must be evaluated.

Each multi-agent goal network outputs the probability that a particular goal is active at each time in the play. However, the system still needs to compare between particular plays as the recognition proceeds. By comparing the output of these networks, we believe that the system can find the most probable play description. Even if the probability of the correct play is relatively low, we expect that the correct network will be "swamped with the truth" and have a disproportionately high value compared with other networks. Because we have not yet completed a significant number of multi-player networks we cannot evaluate this approach.

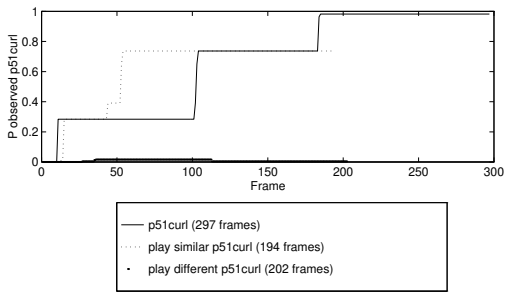

Figure 5: Result of evaluating a simple version of the p51curl multi-agent network on three different plays.

4.3 Status

Currently we have about 80 plays manually tracked that we are using for testing our visual networks and multi-agent networks. The system has knowledge about 40 region definitions, 60 object types, and `ISA` relationships between objects. We have constructed approximately 60 evidence detectors, most of which work on individual agents, arbitrary groups of agents, and sometimes regions and which output fuzzy quantifications of information computed based on trajectories. We estimate 70 visual networks will ultimately be required for recognition of most of the plays in our database, and about a third of those have been constructed.

Figure 5 shows a preliminary result with for the p51curl play using only 3 agents and 7 goals. The real p51curl data receives the highest probability at the end of the observed play. A p51curl-like play receives a lower probability, and a completely different play has a very low probability. We are currently constructing more visual networks and tuning the probability assignment in the multi-agent networks so the system can be tested on complete play descriptions.

5 Prior work

There has been little prior work describing frameworks for real systems that recognize multi-object action with approximate models and noisy input signals. However, several systems have components that we have integrated in our work. Most multi-agent plan recognition work focuses on modal logics that can represent joint intentionality but which have not been implemented in realistic domains with noisy evidence detectors [11, 7, 26]. Until the introduction of probabilistic plan networks, the dominant plan representation model was plan hierarchies [18]. Using Bayesian networks to probabilistically model plans relaxed the strict assumptions of the plan hierarchies model. Networks can represent multiple top-level goals where probabilistic priors can be used to rank two equally possible but not equally likely plans [6]. Further, they have been used to integrate "action patterns" and beliefs about agent's mental state [25]. Dynamic belief networks (DBNs) [30], which include hidden Markov models (HMMs), have been used with some success but have not been demonstrated to be appropriate for domains where temporal relationships are not easily encoded as first-order Markovian. This work, like that of Hanks, Madigan, and Gavrin, assumes that temporal information is most naturally extracted from the expert, not obtained from the data. Therefore DBN representations that use time series propagation of temporal properties are not necessary the most appropriate model[12].

Huber has shown that simple goal recognition belief networks can be constructed automatically from representations of action used by a plan generation system and then used by a planning agent in a multi-object scene [14]. This project builds on Huber's work of automatic construction of networks. Existing systems for probabilistic plan recognition have not been demonstrated on problems where it is necessary to represent fuzzy temporal relationships among multiple goals of multiple agents. Search-based systems designed specifically to recognize multi-agent goals and actions outside of probabilistic frameworks [28, 3, 31] are sensitive to noisy data and detectors.

Belief networks have been used for visual recognition of static objects [20, 1] and for visual attention selection[29]. Promising work on recognizing single-agent action from trajectory information using transition diagrams and fuzzy reasoning [19, 23] led us to investigate the use of belief networks, which more explicitly represent knowledge dependencies and are computationally well-understood, for multi-agent action recognition. Previous work in traffic understanding has used an agent-based belief network and agent-centered features for recognition of driving activity from simulated data[8] and real data[5, 13]. Unlike this work and other systems that reason about action using physics-based properties[21], here the system must also represent and recognize high-level domain knowledge requiring that logical and temporal relationships between multiple agents be modeled. Remagnino, Tan, and Baker recently described a pedestrian and car tracking and surveillance system that models the interaction between any two agents using a small belief network[27]. Here the simultaneous behavior of up to 11 agents interacting with more complex temporal behavior must be modeled and recognized.

6 Final remarks

We have proposed a representation – motivated by findings in the computer vision object recognition literature – that extends previous work in high-level visual motion recognition to handle multi-agent plans with complex temporal and spatial interactions among objects. We represent visual action using temporal graphs. The primitives in these graphs are agent-based belief networks that can recognize agent goals by probabilistic integration of visual evidence. Our networks, which are small and therefore computationally manageable, contain nodes that encode binary spatial and temporal relationships. We have demonstrated that these networks can recognize single agent and multi-agent action when a probabilistic description of the multi-agent action is automatically generated from a simple description of a team plan. Studying the representational, recognition, and computational properties of the multi-agent networks is the focus of our current work.

References

[1] J.M. Agosta. The structure of Bayes networks for visual recognition. In *Uncertainty in Artificial In-*

telligence, pages 397–405. Elsevier Science Publishers B.V., North-Holland, 1990.

[2] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 1983.

[3] J. Azarewicz, G. Fala, and C. Heithecker. Template-based multi-agent plan recognition for tactical situation assessment. In *Proceedings of the sixth conference on artificial intelligence applications*, pages 248–254, 1989.

[4] A.F. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. In *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, February 1997. To appear: Proceedings of the Royal Society.

[5] H. Buxton and S. Gong. Advanced visual surveillance using Bayesian networks. In *Proc. of the Workshop on Context-Based Vision*, 1995.

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

[7] P.R. Cohen and H.J. Levesque. Teamwork. *Nous*, 25(4):487–512, 1991.

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

[9] W.E.L. Grimson. *Object recognition by computer: the role of geometric constraints.* MIT Press, Cambridge, MA, 1990.

[10] 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.

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

[12] S. Hanks, D. Madigan, and J. Garvrin. Probabilistic temporal reasoning with endogenous change. In P. Besnard and S. Hanks, editors, *Int'l Conference on Uncertainty in Artificial Intelligence*, volume 11, San Francisco, CA, August 1995. Morgan Kaufmann Publishers.

[13] 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.

[14] M.J. Huber. *Plan-based plan recognition models for the effective coordination of agents through observation.* PhD thesis, University of Michigan, 1996.

[15] M.J. Huber, E.H. Durfee, and M.P. Wellman. The automated mapping of plans for plan recognition. In *Int'l Conference on Uncertainty in Artificial Intelligence*, pages 344–351, July 1994.

[16] S.S. Intille and A.F. Bobick. Closed-world tracking. In *Proc. Int'l Conf. Computer Vision*, pages 672–678, June 1995.

[17] F.V. Jensen. *An introduction to Bayesian networks.* Springer-Verlag, 1996.

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

[19] H. Kollnig, H.-H. Nagel, and M. Otte. Association of motion verbs with vehcile movements extracted from dense optical flow fields. In J.-O. Eklundh, editor, *Proc. European Conf. Computer Vision*, volume 2, pages 338–347. Springer-Verlag, 1994.

[20] T.S. Levitt, J.M. Agosta, and T.O. Binford. Model-based influence diagrams for machine vision. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 5, pages 371–388. Elsevier Science Publishers B.V. (North Holland), 1990.

[21] R. Mann, A. Jepson, and J.M. Siskind. Computational perception of scene dynamics. In *Proc. European Conf. Computer Vision*, pages 528–539, April 1996.

[22] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Comp.*, 6(2):59–74, 1988.

[23] 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.

[24] J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[25] 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.

[26] A.S. Rao and G. Murray. Multi-agent mental-state recognition and its application to air combat modeling. In *Proc. Work. Distributed AI*, July 1994.

[27] 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.

[28] 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.

[29] R.D. Rimey and C.M. Brown. Where to look next using a Bayes net: incorporating geometric relations. In *Proc. European Conf. Computer Vision*, 1992.

[30] S. Russell and P. Norvig. *Artificial intelligence: a modern approach.* Prentice Hall, Inc., Upper Saddle River, NJ 07458, 1995.

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

[32] M. Tambe. Agent architectures for flexible, practical teamwork. In *Proc. Nat. Conf. on Artificial Intelligence*, 1997.

v