Propagation Networks for Recognition of Partially Ordered Sequential Action

Yifan Shi, Yan Huang, David Minnen, Aaron Bobick, Irfan Essa
GVU Center / College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
{monsoon, huangy, dminn, afb, irfan}@cc.gatech.edu

Abstract

We present Propagation Networks (P-Nets), a novel approach for representing and recognizing sequential activities that include parallel streams of action. We represent each activity using partially ordered intervals. Each interval is restricted by both temporal and logical constraints, including information about its duration and its temporal relationship with other intervals. P-Nets associate one node with each temporal interval. Each node is triggered according to a probability density function that depends on the state of its parent nodes. Each node also has an associated observation function that characterizes supporting perceptual evidence. To facilitate real-time analysis, we introduce a particle filter framework to explore the conditional state space. We modify the original Condensation algorithm to more efficiently sample a discrete state space (D-Condensation). Experiments in the domain of blood glucose monitor calibration demonstrate both the representational power of P-Nets and the effectiveness of the D-Condensation algorithm.

1. Introduction

Automated recognition of daily activities can provide the contextual information necessary to implement a wide range of assistive technologies, smart appliances, and aware environments. To this end, there has been extensive research in developing systems that recognize, annotate, or respond to the activity of a user (*e.g.*, [1, 3, 4, 7]). Most of these approaches consider activity as a temporally ordered single stream of *instantaneous* events. The underlying representations are typically finite state machines (FSMs) (either deterministic [6] or probabilistic [17]) or an extension such as context-free grammars [9, 11, 12]. Detected events cause transitions in the graph and a successful transition through the entire graph implies the recognition of the represented activity.

In this paper we present an alternate approach. First, we presume that elemental or primitive *intervals* are the basic units that are sequenced to define higher level activities. Second, we assume that there are temporal and logical constraints that can enforce triggering relationships between actions. Take, for example, the activity of reading a book, which we might characterize as follows. To *read* a book, a person needs to *retrieve* the book, *open* it, *look* at it for some length of time, and then *close* the book and *return* it to the shelf. Each of these steps has temporal extent, and some intervals occur in parallel. For instance, during the interval of *looking* at the book, there may be occasional intervals of *flipping* the pages.

We have devised a representational mechanism and interpretation method that explicitly encodes these aspects. We begin by describing the overall framework — a *Propagation Network (P-Net)* — and how it differs from typical graphical model representations in terms of both instantaneous evidence and temporal evolution. Next, we present a discrete particle filter based search algorithm *(D-Condensation)*, that seeks to find an interpretation of the observed activity that maximizes the overall likelihood subject to the encoded constraints. We demonstrate the effectiveness of the P-Net by presenting recognition results on 41 video sequences depicting a person calibrating a blood glucose monitor. Using vision-based hand and object tracking along with state information measured directly from the glucose monitor, the P-Net not only recognizes successful execution of a calibration procedure but also identifies omissions when the user misses a step. Finally, we compare the P-Net approach with previously proposed stochastic context free grammar (SCFG) methods and discuss some important advantages exhibited by P-Nets.

2. Previous Work and Motivation

There has been considerable research exploring how to represent and recognize activity. Here we only mention those efforts that contribute directly to the current proposal.

Starting with Yamato [18] and continuing predominantly in the gesture recognition community (*e.g.*, [17]), researchers have turned to hidden Markov mod-

els (HMMs). The appeal is obvious: HMMs provide solutions to the representation, recognition, and learning problems [15]. There are several difficulties with this approach however, of which the most severe is the complexity of representing concurrent actions. It is not uncommon to have sequenced primitives with parallel tracks, each of which needs to be completed before continuing on to some later action. A major difficulty with an HMM or other FSM representation such as those described above or in [6] is that the system can only be in one state at a time, and the transitions across states are instantaneous events. One notable approach to remedy this situation is found in [5], where multiple networks are coordinated.

In an HMM, at each point in time there is a prior density on the state distribution determined by the previous time step, and the likelihood of the current measurement depends only on the current state. This structure is exploited by dynamic Bayesian networks (DBNs) where at each time step the posterior probability at time t becomes the prior probability for time $t + 1$. DBNs have been used to assist tracking and also for decomposing sequences into their independent processes [10]. The reasoning mechanism of Propagation Nets proposed here is closely related to DBNs.

Finally, activities are often composed of partially ordered, sometimes parallel, finite duration intervals. For example, "the chef is holding the knife *while* he is chopping ingredients" includes parallel actions. Very few approaches that can express this kind of relationship have appeared in the recognition literature. One exception is the work of Pinhanez [14] that employs a simplified version of Allen's interval algebra to reason about temporal constraints. Within that system one can naturally represent, for example, that two intervals may occur in parallel (or in arbitrary order) and both must complete before a third begins.

Our work is largely motivated by the desire to create assistive technology within a domestic environment. In this paper, we focus on the specific activity of calibrating a blood glucose monitor, a common task for elderly people who develop late stage diabetes. Though these devices are promoted as being easy to use ("only 3 steps"), careful task analysis shows that as many as 52 independent operations are required [16]. In addition, the long series of sequential tasks required for successful blood glucose monitoring are sensitive to procedural errors and may lead to health risks if performed incorrectly [13]. For the activity models in this paper, we use a slightly coarser granularity resulting in 14 identifiable steps. These steps are shown in the conceptual P-Net representation of Figure 1.

3. Representing Sequential Activity

Consider again the example of reading a book. The primitive intervals as well as the temporal relationships include, "first, [A] fetch the book; *next*, [B] look at the book *while* occasionally [C] flipping the pages; *finally*, [D] put down the book." Even this relatively trivial example suggests that we need a variety of relationships to represent activity:

Sequential streams: There is a natural partial ordering of components.

Duration of elements: The primitives are not events but have temporal extent.

Multiple, parallel streams: Many intervals may occur in parallel.

Logical constraints: Some intervals can be satisfied by a disjunction of sub-intervals.

Non-adjacency: Sequenced intervals may not meet but may only be ordered.

Uncertainty of underlying vision component: Extracted features will always be noisy.

Given these observations, we need an activity model that can (1) represent an activity by a collection of elements each of which corresponds to a temporal interval primitive, (2) encode temporal and logical constraints between these elements, and (3) permit efficient computation that determines when a sequence of observations is an example of performing the activity.

3.1. Conceptual overview of Propagation Networks

To incorporate the above characteristics of activity, we propose a new representation schema, Propagation Networks (P-Nets), and a corresponding inference algorithm called D-Condensation.

A P-Net represents an activity by associating one action node in the network with each primitive action in the activity. Two dummy nodes, representing the start and end of the activity, are also included. Links in the network correspond to partial order constraints between pairs of actions. Figure 1 provides an example of a conceptual diagram for the P-Net that represents the blood glucose monitor calibration task.

Due to space limitations and to improve clarity, the formulation given here does not permit cycles in the network. Such cycles correspond to multiple occurrences of the same set of primitive actions. The P-Net formalism, however, does not preclude the representation of

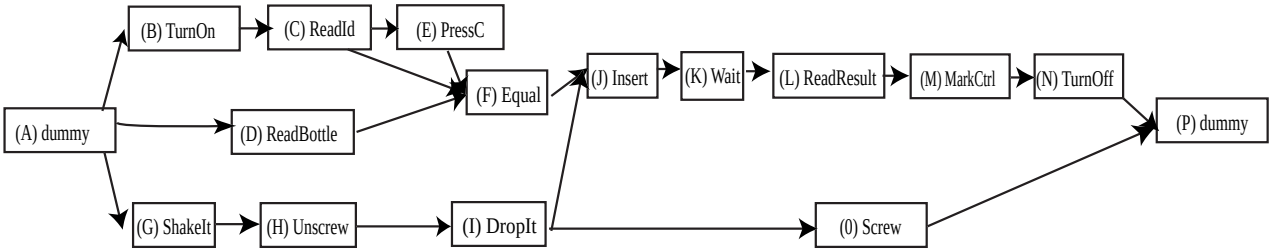

Figure 1: Conceptual diagram for the glucose calibration task

cycles, and only requires a straightforward extension accomplished by using a slightly more complex definition of the directed links.

Each action node includes a duration model. The state of each node, described in detail in the next section, includes the time that the node (most recently) started and its duration. The duration model encodes the probability that the node will remain active during the next time step, based on the node's current duration and the state of its parent nodes.

We enforce the temporal and logical constraints from parents using joint conditional probability functions. Following standard graphical model notation, multiple arrows entering a node implies a joint probability function for the child node conditioned on all of the parent nodes. The joint conditional probability depends on how long the parent nodes were active and how long the child node has been active. Through this conditional distribution, both the temporal constraints and the duration model governing each node are enforced. Note that every node has a self-link that is not explicitly represented in the conceptual diagram of the P-Net.

Finally, each node in a P-Net has an associated evidence component, the behavior of which is characterized by a probabilistic observation model. The evidence component in our experiments is a simple Bayesian network that integrates information from low-level vision detectors (see Figure 3). Though most of the evidence components are instantaneous, they could span a duration such as a backward looking HMM that detects primitive actions [9].

3.2. Computational schema of P-Nets

The actual computation in a P-Net is carried out on a DBN style schema. A P-Net is defined as $\mathcal{P} = \{R, \Phi, B, \Theta, O\}$ as described below.

R: *the random variable set representing the states of the P-Net.* We define the state of each node at time t to be the tuple $\langle s, d\rangle$, where s is the time when the node started and d is the duration that it was (or is) active. Also, we say a node is `nil` if there has been no activation within some history window. Therefore, at time t, the state set for node i is defined as $(\emptyset) \bigcup \langle s, d\rangle$ for $d \geq 0, s + d \leq t$. At each time step, we associate one random variable r_i^t with each action-node and let $P(r_i^t) = P(r_i^t = \langle s, d\rangle)$. The overall state of the P-Net at time t is $R^t = \{r_i^t\}$. Since R^{t-1} d-separates each r_i^t of R^t, we have $P(R^t|R^{t-1}) = \prod P(r_i^t|R^{t-1})$. Finally, since r_i^t is a proper random variable, we have $P(r_i^t = \emptyset) + \sum_{s,d} P(r_i^t = \langle s, d\rangle) = 1$.

Every node in R is either *active* or *inactive*. At time t, any node $r_i^t = \langle s, d\rangle$ is considered active if $s + d = t$ and inactive if $s + d < t$ (note that $s + d > t$ is not possible since the model can not see into the future). Each active node determines its own state in the next time step and can remain active, while inactive nodes serve as parents and conceptually ignite their child nodes. It is this sense of a node being ignited, then finishing, and finally igniting the following node that serves as the origin of the *Propagation Net* name.

Φ: *the causal relationship links in the network.* The causal relationship Φ_i for r_i^{t+1} defines the state transition of a node at each time step. It is defined over the joint set of r_i^t and all r_j^t where j enumerates the parent nodes of i.

In principal, Φ could be quite arbitrary, much like that of DBNs. However, unlike DBNs we include duration modeling in the P-Net structure. This has the possibility of combinatorially exploding the dimensionality of the Φ function. Therefore we constrain this causal function by partitioning it into three mutually exclusive conditions:

1. If any of the parent nodes are active in the previous time step, the child node must remain inactive. This is a simplifying assumption that enforces a staged traversal through the network. We call this situation Φ_1.

2. When the node i is active at time t, (formally $r_i^t = \langle s, t - s\rangle$), Φ_2 depends only on how long the node has been active. That is, once a node is active, only its duration model impacts the likelihood of continuing in the active state.

   Presently, we use Gaussian duration models $\mathcal{N}(u_i, v_i)$ with mean u_i and standard deviation v_i.

We define

$$\Psi_i(d) = (\int_{d+1}^{\infty} N_i(u_i, v_i)/(\int_{d}^{\infty} N_i(u_i, v_i))$$

as the probability that a node will be active for a duration $d+1$ at time $t+1$ if it was active for duration d at time t. Its compliment is the probability that the node will become inactive. Φ_2 is defined by Ψ.

3. The remainder of the Φ PDF is called the activation function. This is the case where node i is not yet active and its parents are no longer active. This "triggering" probability function is restricted to be a function of the time between parent termination and the current time, $t - (s_j + d_j)$ for all parents j. This restriction again reduces the combinatorics of the conditional probability function and allows the system to be trained on a reasonable amount of data.

B: *the observation model.* Every node has an observation model $B(o_i^t) = P(o_i^t|r_i^t)$, which can be discrete or continuous. In the experiments presented below, we chose to use continuous observations with Gaussian distributions. However, there is no fundamental limitation restricting the allowable distributions used in a P-Net. Our current choice for B has the property that all active states of node i conform to one Gaussian distribution, ($\mathcal{N}_i(\mu_{active}, v_{active})$, while all of the inactive states conform to another, $\mathcal{N}_i(\mu_{inactive}, v_{inactive})$.

Θ_i: *the initial distribution of* r_i^0: $\Theta_i = P(r_i^0)$. We usually assume that the P-Net starts in a dummy initial node.

O: *the actual observation sequence,* $o^t = \{o_i^t\}$. The observation sequence is usually an N dimensional sequence as there is one observation per node. Thus, for a P-Net with N nodes and an activity of length T frames, the observation sequence can be represented as an NxT matrix.

4. Activity Recognition with P-Nets

As a tool for activity recognition, we need methods for addressing the following problems:

1. To allow classification of an observation given several P-Nets, calculate the probability of an observation sequence O^T given a P-Net.

2. To determine the times and durations of the actions that compose the activity, compute the most likely internal state sequence given a P-Net.

3. To train the network, compute the most likely P-Net parameters given the observation sequence O^T.

Analogous to the Viterbi algorithm for HMMs, the first two issues can be addressed simultaneously since the probability of the most likely internal sequence can be used to approximate the total observation probability. This state sequence can be computed from:

$$\begin{aligned} P(R^{t+1}|O^{t+1}) &= \textstyle\prod_i P(r_i^{t+1}|O_i^{t+1}) \\ &= c \cdot P(R^t|O^t) \cdot \\ &\quad \textstyle\prod_i P(r_i^{t+1}|r_j^t) P(o_i^{t+1}|r_i^{t+1}), \end{aligned} \tag{1}$$

4.1. D-Condensation

A P-Net specifies the PDF over all of the states at each time step, *i.e.,* $\forall i : p(r_i^{t+1}|r_i^t, r_j^t)$ where j specifies the parents of node i. Although the number of nodes in the P-Net may be relatively small, each node's state set can be quite large. The state set is a discrete distribution over all valid $\langle s_i, d_i \rangle$ pairs, where s can take any value in a fixed temporal window extending backward from the current time. Due to the large size of each state set, standard Bayesian net inference algorithms are infeasible. In this section we introduce Discrete Condensation (D-Condensation) to efficiently search through R^t at each time step.

We represent the possible states of the P-Net as a set of weighted particles. Each particle is comprised of at least one and potentially several tokens, and each token represents one active, parallel action stream. The tokens maintain their history and also store the current node's state (*i.e.,* $\langle s_i, d_i \rangle$). The Condensation algorithm provides one approach for propagating these particles forward through time. However, since in a network with N nodes and a window size of w there are $O(w^{2N})$ possible states, standard Condensation is impractical. In addition, Condensation with importance sampling [8] is precluded by the difficulty of building a viable importance function in this space. The main difficulty for standard particle filters is that they will quickly force all of the particles to be the same or nearly the same as the most likely particle. In addition, the propagation mechanism causes most of the particles to follow very similar paths through the network. This over-clumping of particles has been observed in other research [2]. The implication is that exploration of the state space is very slow, and a huge number of particles is required to explore low probability paths.

We therefore propose D-condensation by taking advantage of the limited branching factor of P-Net states to improve efficiency. For example, when node i is active at time t, $r_i^t = \langle s, t - s \rangle$, the next step must be either $r_i^{t+1} = \langle s, t - s + 1 \rangle$ or $r_i^{t+1} = \langle s, t - s \rangle$. We need not explore other state possibilities. For any particle, we may generate at most $O(2^J)$ subsequent particles, where

J is the size of the largest cut set in the P-Net that separates the dummy *start* node from the dummy *end* node. So for M particles, at most $M \cdot 2^J$ calculations are performed. To further reduce the computation load we use a beam search and merge all particles with the same state by removing all but the most probable. Though this upper bound is still exponential in the worst case, in practice the maximum number of surviving particles is small. Experimental results are presented in Section 5 that verify this claim and, in fact, show faster than real-time performance.

According to the observation model, even if there is no active node, there is still a small probability, L, that any current observation can be seen. Since we only care about comparing particles, L can be deemed a constant and divided out in each time step. Equation 1 can thus be further simplified as:

$$P(R^{t+1}|O^{t+1}) = c \cdot L \cdot P(R^t|O^t) \cdot \prod P(r_i^{t+1}|r_j^t) P(o_i^{t+1}|r_i^{t+1}) / P(O_i^{t+1}|\emptyset), \quad (2)$$

where node i is active, c is a normalization constant and $L = \prod P(o_i^{t+1}|\emptyset)$.

In this manner, we can iteratively obtain the distribution of $P(R|O)$. And, at the end of the activity sequence, the best interpretation will correspond to the history path of the most likely particle.

4.2. Training

Three elements of the P-Net can be learned from training data: the Gaussian duration model, the inter-link probability, and the observation model. The temporal and logical relationships between nodes, however, are prescribed by a knowledge engineer when the model is designed.

To initialize the training, we use manually labeled data sequences to estimate the observation model and the Gaussian duration model. Then we use a standard EM algorithm to improve these estimates using unlabeled data.

To facilitate training on a relatively small number of unlabeled data sequences, we further decompose the Φ function into noisy-and and noisy-or functions. This implies independence between parent nodes and allows each link to be trained separately.

For each training sequence, the highest probability internal state sequence is computed using the initial P-Net. Then, the observation segment $o_{(i,k_1)}, ..., o_{(i,k_2)}$ for node i which corresponds to the activation of the node according to the internal state sequence is used to update the estimates for the duration model, inter-link probability, and observation model.

4.3. Classification

In real world applications, people who perform activities sometimes make mistakes. Such mistakes deviate from the ideal activity described by the P-Net in the form of insertion errors, which are unexpected actions, and deletion errors, which are skipped actions. This leads to three possible classifications for an observation sequence: (1) a correct example, (2) an almost correct example with a small number of mistakes, and (3) a negative example. P-Nets are capable of dealing with insertion errors and deletion errors. The inference algorithm will ignore the input that does not contribute to the sequence interpretation with only a small penalty. P-Nets deal with deletion errors by using low probability data to push through the missing nodes. Such hallucinations can be detected by checking the history path. If $\prod P(o_{i,k}|R_i)$ is below a threshold or the activation length is less than another threshold, the node is considered to be hallucinated.

If the best particle is able to reach the final dummy node with a probability below a learned threshold, the sequence is classified as a *negative* example. Otherwise, if a further check through the history of the particle reveals no hallucinations, then it labeled as a *correct* sequence. If hallucinations do exist, then it is considered *almost correct*. Thus, P-Nets can not only label the whole sequence as one of three categories, but also label each frame and automatically detect any missing nodes.

5. Experiments

We present our experiments with the glucose monitor calibration task as define by the P-Net depicted in Figure 1. As mentioned, this domain is of current interest as we explore the application of assistive technology for elderly care.

5.1. System architecture

The system is constructed in a layered framework as shown in Figure 3. At the bottom most layer is an input stream of raw sensor information, in our case both video frames and an RS232 stream from the glucose monitor itself. The RS232 stream provides basic information about the state of the glucose monitor which is also availabe visually from the device's screen. A tracking module, described in the next section, provides (x,y) location and relative distances between objects. The tracking information and the device state serve as input to Bayesian networks that assert instantaneous primitives such as unscrewing the cap or reading results from the monitor's screen. These Bayesian networks serve as the observation models for the P-Net, which is represented at the top of the system architecture diagram.

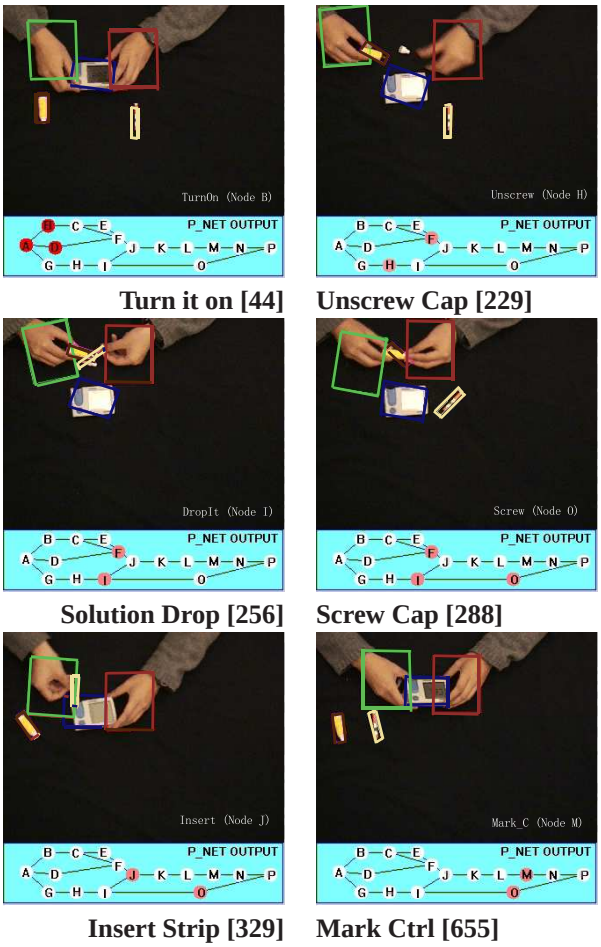

Figure 2: One of our test data sequences with the P-Net shown below it for various actions. [·] shows the frame number in the sequence. The color of the action nodes in P-Net shows P-Net belief on whether the action is occurring. (Refer to Figure 1 for the actions represented by the nodes.)

5.2. Tracking and observation data

To provide visual input, we constructed an indoor vision tracking system that uses particle filters to track multiple objects including hands, the testing strip, the liquid bottle, and the glucose monitor.

We create one tracker for each object and randomly initialize its particle locations. Each particle is represented by a 3-vector, $x_t = (t_x, t_y, r_\theta,)$, where t_x and t_y are the translation along x and y directions respectively, and θ is the angle of rotation in the image plane. Two statistical features, color histograms and orientation histograms, are used to measure the similarity between the image and the template corresponding to the particle state and thus allow computation of the particle likelihood. Both of these features are computationally simple and insensitive to variations in image scaling and rotation. Figure 2 gives some tracked key frames of the glucose monitoring sequence with each key frame representing one salient event node in the P-Net.

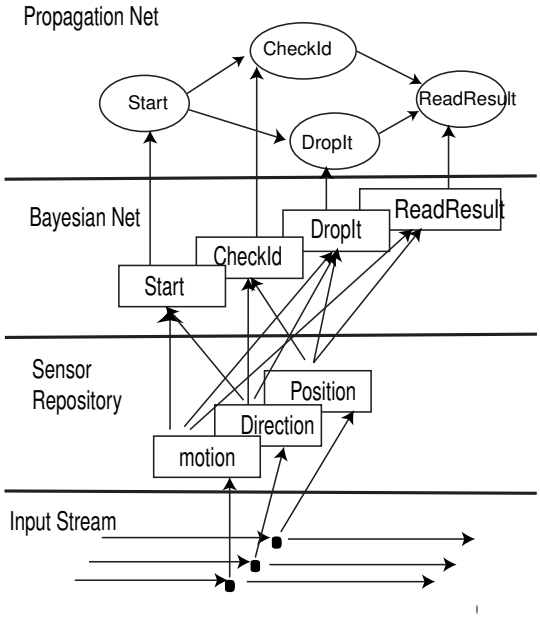

Figure 3: System architecture

5.3. Modeling glucose monitor calibration

We built a 16 node P-Net representation for the standard glucose monitor calibration procedure. Then we had three subjects perform a total of 21 correct sequences, 10 missing-one-step sequences and 10 missing-six-steps sequences. For training, we used six correct sequences, and saved the rest for testing.

The middle level output from the Bayesian networks is quite poor, as the low-level detectors generate too many false alarms. The temporal constraints encoded in the P-Net, however, cause the final labeling to be much better. An example comparing the `shaking` action is shown in Figure 4.

In our experiments, D-Condensation never generates more than 1,967 particles and the number of unique active particles never exceeds 238. Considering every node has a temporal window length of at least 50, which makes the number of possible states for each node at least 1,250, we see that the D-Condensation algorithm uses a relatively small number of particles to explore the state space. D-Condensation is very fast. The overall frame rate on pre-processed observation data (*i.e.*, on the output of the Bayesian network layer) is over 122 frames per second. This is faster than real time and more than sufficient for real world applications. The computational statistics are summarized in Table 3.

The final results are shown in Table 1. All correct sequences are recognized. Eight out of the 10 missing-one-

Table 1: Overall evaluation

Sequence Category	Total	Correct	Almost Right	Negative
Training	6	100%	0%	0%
Correct	15	100%	0%	0%
Missing One	10	20%	80%†	0%
Missing Six	10	0%	50%‡	50%

† All 8 claim missing that step; 2 of 8 claim missing an extra step; 1 claims missing extra 2.

‡ 3 claim missing 5 nodes, 2 claim missing 6; all 5 at least claim 3 actual missing steps.

Table 2: Labeling individual nodes

Individual Node	Overall Success †	Correct Positive ‡	Correct Negative ∗
B:TurnOn	0.9999	1.0000	0.9999
C:RdIdScreen	0.9901	0.9956	0.9897
D:RdIdSstrip	0.9893	0.9333	0.9909
E:PressC	0.9787	0.2344	0.9998
F:Equal	0.9847	0.9267	0.9908
G:ShakeIt	0.9590	0.6003	0.9738
H:Unscrew	0.9563	0.5041	0.9857
I:DropIt	0.9827	0.8584	0.9941
J:Insert	0.9878	0.8643	0.9961
K:Wait	0.9964	0.9987	0.9958
L:ReadResult	0.9966	0.9847	0.9991
M:MarkCtrl	0.9983	0.9720	0.9993
N:TurnOff	0.9967	0.8997	0.9997
O:screw	0.9476	0.6629	0.9617
Average	0.9839	0.8709	0.9914

† Overall Success is the average of all nodes.

‡ Correct Positive:number of correctly labeled positive frames over number of all positive frames for node i

∗ Correct Negative:number of correctly labeled negative frames over number of all negative frames for node i

step sequences are identified, while the other two are labelled as correct. Checking back with ground truth, we find that the mistakes are caused by insertion errors in the vision module and that the insertion errors make the sequences statistically indistinguishable from correct sequences. We label five of the missing-six-steps testing sequences as totally wrong and label five as almost correct.

The results can also be evaluated by labelling each frame. This provides 16 binary values per frame, each specifying whether a particular node in the P-Net is active or inactive. By comparing these labels with ground truth, we can compute the overall-correct-ratio as *[correct_positive + correct_negative]/[all_frames]*, the correct-positive-ratio as *[correct_positive]/[all_positive]*, and the correct-negative-ratio as *[correct_negative]/[all_negative]*. Such statistics for the test sequences are available in Table 2. Though the individual labelling ratios have a fairly wide range, the overall correct ratio for any sequence is very high (over 98%), and the average correct-positive-ratio is also over 87%.

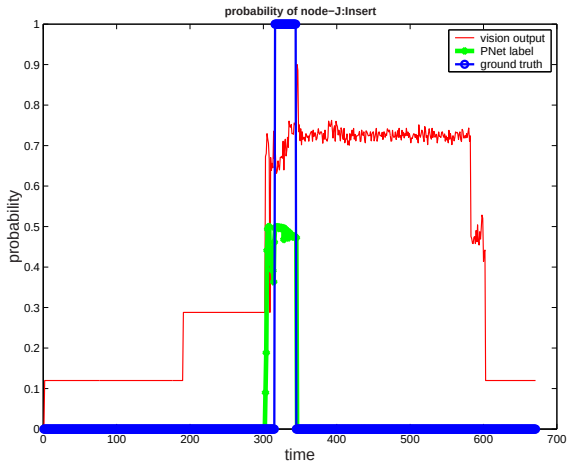

Figure 4: Comparison of P-Net labels with ground truth and low-level vision output

Table 3: Overall computational performance

Measure	Data
Sequence Length Range	[232, 928]
Average Speed (frames/sec)	122.7
Maximal Distinctive Particles	238
Maximal Subsequent States	1967

5.4. Comparison to SCFGs

To provide a comparison to P-Nets, we created a model of the blood glucose calibration task using a stochastic context-free grammar [9, 12, 11]. We associated one event in the grammar with every node in the P-Net. Events were detected by finding intervals of high probability in the corresponding low-level observation probability signal, $p(O_i)$. For each interval, an event was generated every 45 time steps and inserted into the event stream.

A stochastic grammar must explicitly represent all valid event orders. For this experiment, we enumerated all 1,624 event sequences implicitly represented by the P-Net. They were encoded in a stochastic grammar as a single rule with many equally likely production alternatives.

The stochastic parser found valid parses for all 21 of the correct sequences. We measured accuracy by calculating the percentage of symbols in the most likely parse that fell within the correct range according to the ground

truth data. Over the 21 sequences, 62.8% of the symbols were within the correct range. We attribute this relatively low performance to the lack of duration models within the grammar, which causes the parser to be more susceptible to noisy events and to accept temporally unrealistic interpretations.

Computational complexity concerns made parsing the erroneous sequences difficult. Unlike a P-Net, recovering from deletion errors in a stochastic parser incurs an exponential penalty. Thus, parsing the missing-six data set was infeasible. The parser was able to find valid parses for all 10 missing-one sequences, but only after a restricted grammar (35 alternatives) was used and deletion recovery was limited to only one consecutive event hallucination.

6. Conclusion

The P-Net and the associated D-Condensation algorithm provide a natural and efficient way to integrate temporal and logical relationships in daily activity. Experiments show that they are robust and efficient with regard to real activities, even in the presence of insertion and deletion errors.

Our architecture not only provides an activity recognition method, but also a real-time control method. It can tell what is happening and also what is expected to happen. This is a very powerful tool for focusing a vision system on highly interesting areas to extract relevant information.

In conclusion, P-Nets have two major advantages over traditional techniques. They can represent an activity with parallel action streams and provide a better model of the world by representing temporal intervals rather than instantaneous events.

References

[1] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", *IEEE Tran. Patt. Anal. and Machine Intelligence* **21**(11), pp. 1241-1247 (1999);

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking." *IEEE Transactions on Signal Processing*, **50**(2), pp174–188, 2002.

[3] A.F. Bobick and J. Davis, "The Recognition of Human Movement Using Temporal Templates ",*IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, pp257-267, 2001.

[4] Vaswani, N. and Roy Chowdhury, A. and Chellappa, R."Activity Recognition Using the Dynamics of the Configuration of Interacting Objects", *CVPR'2003 pp633-640*

[5] Shaogang Gong and Tao Xiang, "Recognition of Group Activities using a Dynamic Probabilistic Network". In Proceedings of IEEE ICCV 2003 Conference, pp 742-749, 2003.

[6] Hongeng, S. and Nevatia, R. "Multi-Agent Event Recognition", ICCV'2001, PP84-91

[7] Grimson, W.E.L. and Lee, L. and Romano, R. and Stauffer, C., "Using Adaptive Tracking to Classify and Monitor Activities in a Site", CVPR'1998, pp22-31.

[8] M. Isard and A. Blake. "Icondensation: Unifying low-level and high-level tracking in a stochastic framework", pp893–908, ECCV'1998

[9] Y. Ivanov and A. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing", *IEEE Tran. PAMI*, **22**(8), August 2000

[10] D. Koller and J. Weber, "Towards Robust Automatic Traffic Scene Analysis in Real-Time", *ICPR'1994*, pp. 126-131, 1994.

[11] D. Minnen, I. Essa and T. Starner. Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, Madision, Wisconsin, 2002.

[12] D.J. Moore and I. Essa. Recognizing multitasked activities using stochastic context-free grammar from video. In *Proceedings of AAAI Conference*, 2002.

[13] A. L.Mykityshyn, A. D. Fisk and W. A. Rogers, "Learning to use a home medical device: Mediating age-related differences with training." In *Human Factors*, 44, pp. 354-364, 2002.

[14] C. Pinhanez, "Representation and recognition of Action in interactive Spaces", Ph.D thesis, MIT Media Lab, 1999.

[15] L. R. Rabiner and B. H. Juang, *Fundamentals of speech recognition*, Prentice Hall, 1993.

[16] W. A. Rogers, A. L. Mykityshyn, R. H. Campbell and A. D. Fisk, "Analysis of a simple medical device." In *Ergonomics in Design*, 9, 6-14. 2001.

[17] T. Starner and A. Pentland, "Visual recognition of American Sign Language using hidden Markov models," *Intl. Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[18] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," *CVPR*, 1992.