# The Representation and Recognition of Activity Using Propagation Nets

Yifan Shi    Aaron F. Bobick

GVU Center / College of Computing
Georgia Tech
Atlanta, GA 30332
{monsoon|afb}@cc.gatech.edu

## Abstract

*In this paper, we present a new mechanism for the representation and recognition of sequential action. The underlying premise is that many activities can be represented by a partially ordered set of intervals each corresponding to a primitive or elemental action. Each interval typically has associated parent intervals whose states determine the likelihood that the child interval will be triggered. Furthermore, each interval typically has perceptual evidence that is indicative of the elemental action taking place. We present a temporal sequencing algorithm that attempts to interpret an multi-dimensional observation sequence of visual evidence as a temporal propagation through a network of these intervals. We develop, implement, and test a particular mechanism for embodying this representation within the domain of a small number of indoor human activities.*

**Keywords:** Activity recognition, Bayesian network, finite state ma- chine, stochastic state propagation

## 1. Introduction

Rich vision interfaces react to more than particular instantaneous acts by the user. Rather, the system needs to be sensitive to more temporally extended patterns of activity. The simplest characterization of these systems is that they need to be aware of what the user is doing in terms of some set of understood activities.

There has been extensive work recently in developing perception systems that respond to the action of a user. A few timely reviews can be found in [2, 9]. Most of these approaches consider activity as a temporally order sequence of *instantaneous events*. The underlying representations are typically finite state machines (either deterministic[4] or probabilistic[14]) or some extension such as context-free grammars [5, 10]. The 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 take a somewhat different approach. First, we presume that elemental or primitive *intervals* make up the basic units that are sequenced to define higher level activities. Second, we assume that there is a temporal/logical relationship that can be loosely thought of as a partially ordered set. An example might be that to make a phone call (on a phone with a separate handset from dialing base) the subject needs to pick up the phone, then accomplish both elemental actions of dialing and putting the phone to his ear, and then speaking into the phone. The dialing and lifting to the ear both happen after picking up the handset, and both must occur primarily before speaking, but there is no hard relationship between them. And third, we associate some form of perceptual evidence with each of the underlying intervals.

In this paper we devise a representational mechanism and interpretation method that explicitly encodes these three aspects. We begin by describing the overall framework a *propagation network* and how it differs from typical graphical model representations in terms of both instantaneous evidence and temporal evolution. Next we will present a approximate algorithm — local maximal search algorithm (LMSA) — that seeks to maximize the overall evaluation based on positive evidence that an interval will start or continue. Finally we conduct some experiments using motion capture data as input for the recognition of some simple office-relevant activities such as making a phone call or reading a book.

## 2. Representing sequential activity

As mentioned, there has been extensive work in representing and recognizing activity. Here we only mention those efforts that contribute to the current proposal. We then separate out the elements of perceptual evidence from those of historical (really temporally contextual) support.

## 2.1. Previous work

Starting with Yamato [16] and continuing predominantly in the gesture recognition community (e.g. [14]), researchers have turned to hidden Markov models. The appeal is obvious: HMMs provide solutions to the representation, recognition and *learning* problems. Given a set of sequences whose observation are in some feature space, the various HMM algorithms learn discrete states in that space and probabilistic transitions between them, thus constructing a stochastic finite state machine.

The difficulties with this approach are several but we mention only two. First, one problem is the 'H' in the sense that the underlying states are "hidden". For much of activity recognition, the goal is to be able to define higher level activities in terms of understood lower level primitives, making the "hidden" aspect unappealing. Second, it is not uncommon to have partially ordered primitives with parallel tracks, each of which needs to be completed before getting to some end goal. This logicalsequencing constraint is not easily represented by an HMM or FSM.

The relevance of HMMs to the discussion here is that a valuable conceptualization of an HMM is as a conditional-dependence graphical model ( Bayesian Net) unrolled out in time. At each time step $t$ there is a state node $q_t$ that has a probability that it is in each possible HMM state. As a first-order Markovian graphical model, the state probabilities are conditioned upon only the state assignments at the immediately preceding time step. In addition, there is an evidence observation that depends only upon the current state. Once an entire observation sequence is available, the forward-backward or Viterbi algorithm can be evaluated to determine the likelihood of each state at each time, and also the likelihood that the given HMM would generate the observed sequence. This last aspect is what gives HMMs such appeal for activity classification.

In the HMM graphical model at each point in time there is a prior density on state distribution determined by the previous time step, and that the likelihood of the current measurement depends only on the current state. This structure is exploited by Dynamic Bayesian nets (DBN) [8] 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[7].

A major difficulty with finite state machine representation such as those described above or those of [4] is that the system can only be in one state at a time and that transitions across states are instantaneous events. In reality, activities are often comprised of partially order, sometimes parallel finite duration intervals. Very few approaches to representing action this way have appeared in the recognition literature. One exception is the work of Pinhanez [11] that employs a simplified version of Allen's temporal algebra [1] 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) but that both must complete before a third is started.

## 2.2. P-nets: partial sequencing of components

Consider the simple example of making a phone call (on a phone with a separate handset from dial mechanism). A description in terms of primitive intervals might be something like "First, [A] pick up the receiver. Next [B] put the phone to your head AND [C] dial the number (order unimportant, can be parallel). Finally, [D] start speaking." From even this trivial example we see a variety of constraints on any reasonable representation of activity designed for visual recognition:

1. Sequential streams - there is a natural partial ordering of components
2. Multiple, parallel streams - a variety of intervals may occur in parallel
3. Logical constraints
4. Duration of the elements - the primitive are not events but intervals of duration
5. Non-adjacency - sequenced intervals may not meet but only be ordered
6. Uncertainty of underlying vision component - there is always noise in feature extraction and assessment

To address these issues we propose a three tiered representation of activity, the top level of which refers to as a *propagation net* or p-net. (A toy example is shown in Figure 1.) The nodes of a p-net are intended to be primitive intervals — we will consider what it means to be primitive shortly. Loosely we can say that an interval can be said to have some probability of being active at time $t$ having started some duration $d$ before $t$.

The arrows in the p-net are intended to be conditional triggering probabilities. Just as in a DB, the arrows into a node represent a conditional probability on the state of the node at time $t + 1$ given the parents state at time $t$. These conditional probabilities encode the logical requirements, such as that all parents must be active before a child can become active, or that one and only one of possible parents be active for the child to be triggered. In the implementation described here, these conditional

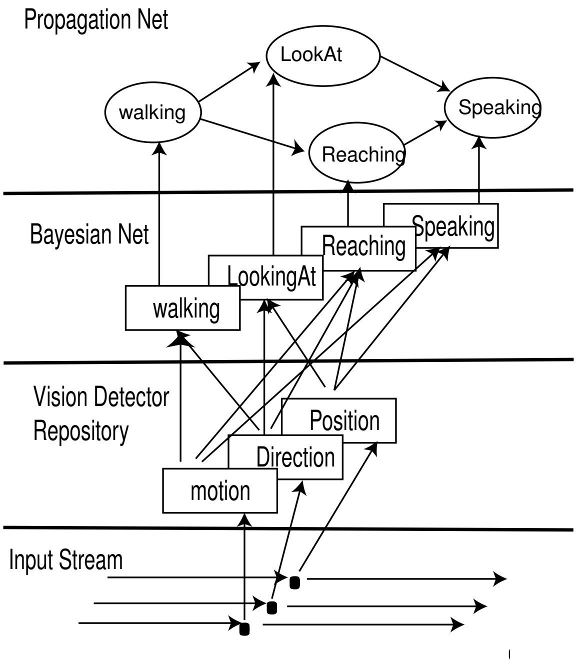


**Figure 1:** Overall architecture. Propagation net is responsible for representing sequencing and logical constraints.

probabilities will specify how likely the child will start at different time after the parent(s) end(s). With multiple parents this probability function can enforce relative timing requirements between the parents should such requirements exist.

An aspect of a p-net that is different than a standard DBN or HMM is that there is a duration model. Namely, for each node there is a duration probability that it will remain active from $t,\ d$ to $t{+}1,\ d{+}1$, and that probability is a function of the $d$ associated with the state at time $t$. Duration-model HMMs are possible at the penalty of increase computational complexity [13].

Each node in a p-net has a corresponding evidence component. The evidence component here is a simple Bayes net that integrates instantaneous information from low-level vision detectors. The repository of detectors is the lowest level of the representation. Though in the development given here the evidence components are instantaneous, they could span a duration such as a backward looking HMM that detects primitive actions [5].

There are some conceptually defined criteria for selecting the detector:

- repetition of the module with some variable parameters
- atomic or self contained vision method

Targeting daily indoor activity, based on such criterion, the following detectors are justified: the motion of the hand, the contact of 2 objects, the orientation of object. Those detectors will provide the probability that the corresponding event is detected in the stream up to the current frame. The output serves are fed into the leaf nodes in the Bayesian network.

Bayesian networks excel in combining inaccurate information and making systematic predictions. We choose them to combine the lower level probability while enforcing the target and object conformity. The definition and calculation for the Bayesian net is standard [6].

# 3. Formal model for p-nets

## 3.1. Definition

Now we provide the precise definition for p-net. We borrow some notation from HMMs in terms of the name of the various elements. p-net is defined as $\mathcal{P} = \{O, B, C, \Phi, S\}$ where:

$S$ is the state set of the nodes. At each time t, for each node i, there will be $d_i$ states. Each $s_i(t,d)$ describes the hypothesis that node i has been active since $t-d+1$ so that duration is $d$. Particularly, $s_i(t,1)$ is influenced by the parent nodes state and $\phi$ (defined below) while $s_i(t, d>1)$ is controlled by $s_i(t, d-1)$ and self-link C. A dummy node is associated with all nodes that don't have predecessor and another is associated with nodes that don't have a successor. They are used to delineate the starting and ending of the whole sequence.

$O = \{o_i(t)\}$ is the observation of node $n_i$ at time $t$.

$B = \{< b_i^+, b_i^- >\}$ is the set of conditional probabilities of observing the perceptual evidence that $n_i$is active depending upon whether the node really is ($b^+$) or is not ($b^-$) active. For continuous output detectors each of the $< b_i^+, b_i^- >$ would have to be a function that can be applied depending upon the observed output value.

$C$ is the conditional probability that describes the self-termination link. It is the conditional probability that given $n_i$ has been activated for $d$ time steps, $n_i$ will become deactivated in the next time step. We model the duration on each node as a Gaussian distribution $G(e_i, f_i)$. Therefore the loopback probability for ni(j) is

$$c_i(d) = \int_d^{d+1} G(e_i, f_i) / \int_d^{\infty} G(e_i, f_i)$$

$\Phi$ defines the set of triggering functions that provide the conditional linking in the node set. For every node $i$, there is a triggering function $\phi_i$ which takes as it's argument the state at time $t$ of the parents of node $n_i$. It returns a probability that node $n_i$ will trigger at $t+1$. As mentioned, for this implementation, the trigger functions rely on the termination of the parent intervals. Thus the probability that a particular node will trigger at $t+1$ depends upon probability of the parent nodes having been active since $t-1-h$, and becoming inactive at time $t$.

In our system, the functions $\phi_i$ are the probabilistic equivalents to logical combinations such as Noisy-OR or Noise-AND [6]. Furthermore, we force the functions $\phi_i$ to be windowed gating functions. It enforces a window on the state history of parent nodes. Only the states which starts within the past h time steps are influential on the child's state.

## 3.2. Local maximal search algorithm (LMSA)

At each time step we need to propagate node activity through the network. We follow the method of DBNs for each state updating. We first define a prior probability on the activity state of node $n_i$ at time $t$ using the state of the node or its parents at time $t-1$ as given and do the conditional propagation. Next we observe the data $o_i(t)$ and combine the prior with the data to get a posterior estimate on the probability of node $n_i$ being active at time $t$. We denote the prior and posterior probabilities for $s_i(t,d)$ as $s_i^{(-)}(t,d)$ and $s_i^{(+)}(t,d)$ .

### 3.2.1 Forward calculation

Because we defined the state termination probabilities $c_i$ to be only a function of the current duration of the activity of a node, the predicted state of a currently active node is only a function of that node:

$$s_i^{(-)}(t,d) = s_i^{(+)}(t-1,d-1)\cdot(1-c_i(d-1))\text{when } d>1$$

The more complex situation arises if a node has not been triggered yet. Now we need to know the probability the node will trigger given the state (and history) of its parents. To estimate that value, we need the triggering function $\phi_i$. In the implementation described here, the $\phi_i$ functions are either a Noisy-OR or Noisy-AND (depending upon the logic of the child). Furthermore, to be more accurate we need to integrate over all possible $t$ and $d$ within $h$ for the parents, but that is computationally too expensive. Instead, we take simplification by selecting the maximal value within the window $h$ and use it as the parent node state to get $s_i^{(-)}(t,1)$:

$$\begin{aligned} s_i^{(-)}(t,1) &= \phi(\{max_{t',d'}(s_j^{(+)}(t',d')(1-c_{j_1}(d'))| \\ &\quad \text{for all} s_j \in \text{Parent}(s_i))) \end{aligned}$$

After computing the prior probabilities, we take our observations and now compute the posterior $s_i^{(+)}(t,d)$ according to Bayes rule using the $b_i$ and the observations $o_i(t)$.

$$s_i^{(+)}(t,d) = b^{(-)}(1-s_i^{(-)}(t,d)) + b^{(+)}s_i^{(-)}(t,d)$$

The interpretation path can be generated by keeping records in each calculation step and trace backward from $s_N^{(-)}(t,1)$.

### 3.2.2 Path evaluation

The forward calculation only selects the usual evidence that will help propagation. If the evidence does not fall in the final path, it has no effect on contributing to the final probability. This is not desirable on overall path evaluation. To amend it, we provide an evaluation function on path.

It is defined as the probability gain to observe the evidence at the predicted evidence probability over nothing happens. In this way, if evidence is not included in the path, then the probability to generate it is 0, and the corresponding evaluation is also 0. Therefore, only the evidence included in the path may change the overall evaluation which is the sum of individual evaluation at each step.

The evaluation is performed when generating posterior probability $s_i^{(+)}(t,d)$.

$$E_i(t,d) = o_i(t) - |o_i(t) - s_i^{(-)}(t,d)b^{(+)} + (1-s_i^{(-)}(t,d))b^{(-)}|$$

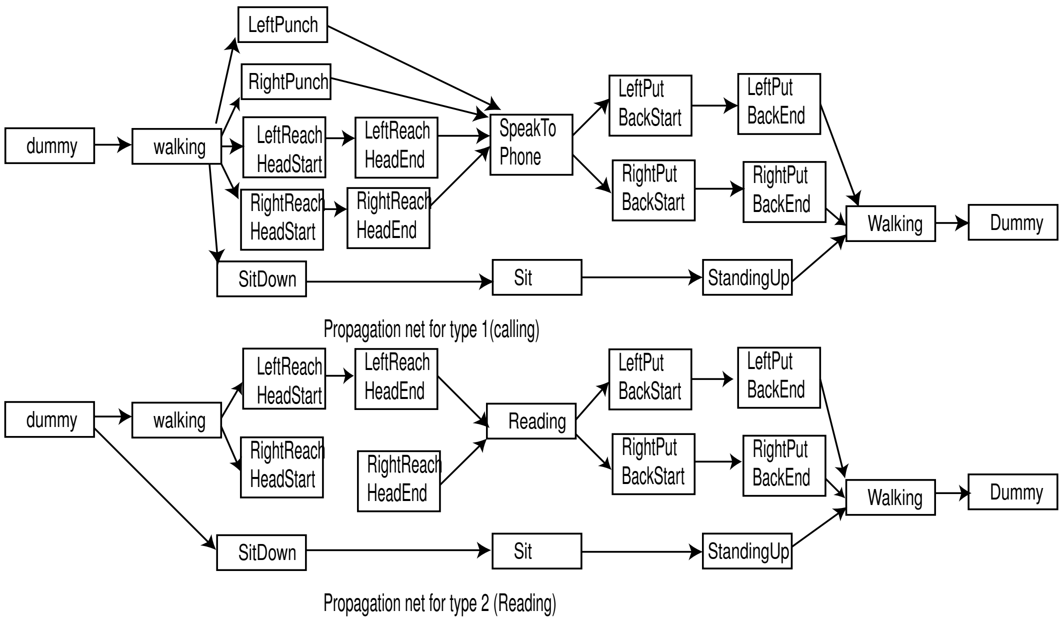
$E(path) = \Sigma_t(Ei(t,d)/N$, for corresponding i, d that falls on path at time t.

### 3.2.3 Iteration

In the forward calculation, the time constraints between the sibling nodes are ignored. As different child nodes may select different parent timeduration state for local maximal, it may lead to a conflicting interpretation. For example, child B may want A to terminate at $t_1$, while child C may want A to terminate at $t_2$. They both assume they succeed. Then the grandson D will obtain a conflicting path from B and C.

To unify the different state assumption on parent node, we choose to iterate on top of the forward calculation. During each iteration, a candidate path is generated backwards from the ending nodes. Then we freeze the node which has no conflicting assumption on itself or on its ancestors. By freeze we mean that, in the next iteration, we restrict the node from generating any states other than the one within the frozen path. And select one assumption for the conflicting nodes which has no conflicting in its ancestor and push the other assumption in stack for later iteration. We then go back to do forward calculation again. Since in each iteration, if there are conflicting nodes, at least one conflicting node doesn't have a conflicting ancestor as they are partially ordered. That node can be frozen in the next iteration. So in each step at least one more node will be frozen so that the iteration will end as the candidate path is limited.

After generating all candidate local maximal paths, we can select the path by its evaluation function. The



Propagation net for type 1(calling)

Propagation net for type 2 (Reading)

**Figure 2:** Results on "calling" and "reading" networks.

path with maximal evaluation is deemed as the overall interpretation.

# 4. Experiments

To test our system, we recorded 2 types of daily activity by motion capture device. In type 1, "calling", one walks in, sit on the chair, make a phone call then walk out; in type 2, "reading", one walks in, sit on the chair , read a few pages of book, then walk out. We took 12 run of each types. Each type is sampled at 10 frames a sec, about 20 seconds long. The propagation net representations for type 1 and 2 are presented in Figure 2.

The parameters for p-net are empirically assigned. The middle level output from Bayesian network is full of noise, as the lower level detector is too simple and has many false alarms. But, the temporal constraints of the p-net cause the mid-level labelling to be much better. An example comparing the `LeftPutbackStart` component is shown in Figure 3.

Finally, Figure 4 demonstrates the initial results of the final output by evaluating all the samples on both networks. As a comparison, we temporally scrambled the data to maintain the same low level evidence but in the wrong order. One can see that the correct interpretation is always the best, though the margins are not as high as needed to do a robust detection. One reason for this is that the two activities are quite similar in motions and thus in terms of low level descriptions. We are currently working on refining the networks.

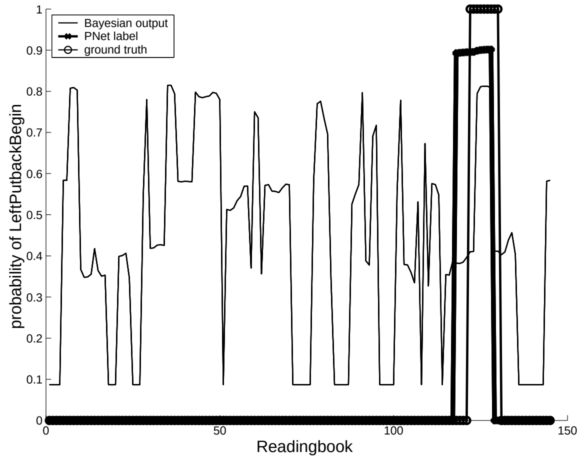


**Figure 3:** Comparison of straight Bayes net detector vs. p-net labelling for `LeftPutbackBegin`.

Since LMSA is based on each node's local maximal, its speed is polynomial. The total running time for a 200 frame sequence is less than 5 seconds on Pentium4 2Ghz. The actual time depends on how many conflicting nodes may exist.

# 5. Conclusion

The PNet and its LMSA algorithm provide a natural and efficient way to integrate temporal and logic relationship in daily activity. Experiments show that they are robust in insertion and deletion error. And the recovery ability is not at the exponential cost.

Our architecture not only provides an activity recognition method, but also a stream labelling method. PNet provides a segmentation on stream and tell what is happening at each time frame. This information is very useful in video stream search. For example, in security mon-

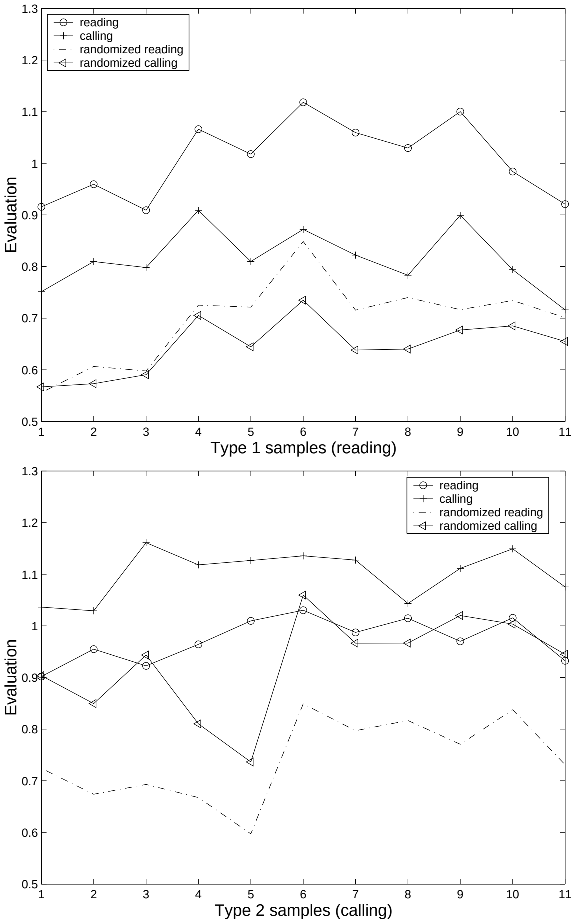


**Figure 4:** Results on “reading” and “calling” networks.

itoring system, critical section of video can be extracted for verification purpose.

There is a major shortcoming on LMSA as it will only produce a result after it see the whole sequence. This restricted the usage from real time control. We are working on improving it.

# References

[1] J.F.Allen, ”Towards a General Theory of Action and Time”, *Artificial Intelligence*, **23**, pp.123-154.1984

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

[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] S. Hongeng, F. Bremond, and R. Nevatia, ”Representation and optimal recognition of human activities”, *CVPR*, 2000

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

[6] F. V. Jensen, *Bayesian Networks and Decision Graphs*, Springer-Verlag, 2001.

[7] N. Jojic, P. Simard, B. Frey, and David Heckerman, “Separating Appearance from Deformation”, *ICCV*, Vancouver, BC, Canada, 2001.

[8] D. Koller and J. Weber, ”Towards Robust Automaic Traffic Scene Analysis in Real-Time”, *ICPR’1994*, pp. 126-131

[9] T. B. Moeslund and E. Granum, ”A survey of computer vision-based Human Motion Capture”, pp. 231- 268, em IJCVIU 2000.

[10] D. Moore, I. Essa, and M. Hayes, “Exploiting Human Actions and Object Context for Recognition Tasks”, *ICCV*, pp. 80–86, 1999.

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

[12] R. Polana and R. Nelson, ”Low Level Recognition Of Human Motion”, *Proc. Of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated objects*, pp. 77-82, 1994.

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

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

[15] N. Vasconcelos and A, Lippman, ”A Bayesian Framework for semantic Content Characterization”, pp566-571, *CVPR*, 1998

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