Recognition of Repetitive Sequential Human Activity

Quanfu Fan, Russell Bobbitt, Yun Zhai, Akira Yanagawa, Sharath Pankanti, Arun Hampapur
IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

Abstract

We present a novel framework for recognizing repetitive sequential events performed by human actors with strong temporal dependencies and potential parallel overlap. Our solution incorporates sub-event (or primitive) detectors and a spatiotemporal model for sequential event changes. We develop an effective and efficient method to integrate primitives into a set of sequential events where strong temporal constraints are imposed on the ordering of the primitives. In particular, the combination process is approached as an optimization problem. A specialized Viterbi algorithm is designed to learn and infer the target sequential events and handle the event overlap simultaneously. To demonstrate the effectiveness of the proposed framework, we report detailed quantitative analysis on a large set of cashier checkout activities in a retail store.

1. Introduction

In this paper, we consider the problem of recognizing repetitive sequential human activities. Such activities are composed of repeated events, each of which is a combination of sub-actions (*primitives*) with certain spatial and temporal constraints. These activities are often observed in workplaces where repeated tasks need to be performed, in which each task consists of a specific set of ordered steps. For instance, in a grocery store, a characteristic sequential action performed by a cashier includes obtaining an item from the lead-in belt, presenting the item to the barcode scanner for pricing and depositing the item onto the take-away belt for bagging (Figure 1). Other real-life examples include product assembly in manufacturing, tailgating and pass check at airports (borders, toll stations or theme parks), just to name a few.

Effective analysis of repetitive sequential activities has broad applications in many contexts, such as workplace safety, fraud detection and product quality assurance. In the example above, fraud happens when the cashier passes an item through the check-out lane without actually registering it in the purchase list. These actions are called fake scans and are referred as *sweethearting* by retailers. The
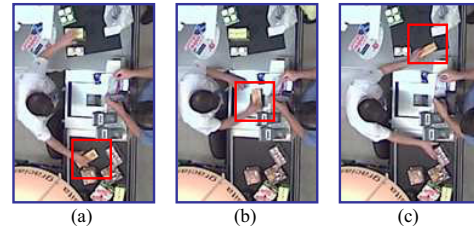

Figure 1. An example visual scan in a retail check-out station, including (a) picking-up, (b) scanning and (c) depositing the item. Visual scan overlap is demonstrated in (a) and (b). Note: Best viewed in color.

term comes from the fact that this type of fraud is performed by cashiers and customers who know each other and act in collusion to obtain free merchandise. *Sweethearting* ranks as one of the most serious problems in the retail industry and causes significant revenue shrinkage with billions of dollars worldwide. We adopt this retail scenario as the embodiment of our proposed framework and use it to demonstrate the effectiveness of our method for solving this real-world problem.

As outlined above, repetitive sequential events exhibit a number of unique characteristics that are distinctive from other types of repeated human actions, such as walking and reading a book. First, in repetitive sequential events, there are strong spatiotemporal constraints imposed on the primitives; the actor typically operates within a relatively confined region and must perform the action quickly. In other types of repetitive actions, such constraints are often loosely enforced. For example, book reading involves repeated page flipping with random actions in between that do not have strong associations. Second, the problem addressed in this paper may involve temporal overlap (e.g., a cashier scans an item while at the same time picking another item up (See Fig. 1c). Such overlap is not possible in other activities, such as walking, where each action must be completed before the next one begins, and thus poses much greater challenges.

We propose a generative framework for recognizing repetitive sequential events, which have strong temporal dependencies and potential parallel overlap, as illustrated in the example above. Our solution combines primitive de-

tectors with a spatiotemporal model for sequential event changes. We consider the recognition task as a process of selecting a set of target sequential events from a large pool of candidates formed by the primitives detected. The selection process is then formulated as an optimization problem where the temporal and spatial constraints between primitives are leveraged. In particular we design a novel Viterbi-like algorithm that can effectively finds an optimal set of sequential events close to the genuine events. The issue of overlap is resolved in the optimization by only searching disjoint sequential events without overlap in the state space. As seen later, this constrained search not only distinguishes our approach from traditional HMMs, but also dramatically reduces the complexity of activity modeling, making the problem tractable. We further apply our proposed framework to recognize the predominant retail cashier activity at the checkout station. We present details on how we model cashier activity in a spatio-temporal way. Finally, we demonstrate the effectiveness of our approach on a large data set captured from a real grocery store including hundreds of transactions and thousands of scanned items.

The paper is organized as follows. In Section 2, we present a literature review on some existing work on human activity recognition; The proposed general repetitive sequential event detection framework is then detailed in Section 3; In Section 4, an application of the proposed framework is demonstrated in the form of an embodiment of recognizing cashier scan actions in a retail environment; Experimental results and performance analysis are presented in Section 5; Finally, Section 6 concludes our work.

2. Related Work

Human activity recognition is a long-studied problem in the field of computer vision. It is obviously impossible to list all existing work, but there are some good literature surveys on this topic, such as the paper by Turaga et.al., [11]. Since human activities are highly spatial and temporally structured entities, a large portion of the existing approaches are based on graphical models, such as Finite State Machines (FSM), Hidden Markov Models (HMM), Context-Free Grammar (CFG) and Dynamic Bayesian Networks (DBN). Mahajan et.al., [5] have proposed a model of multi-layered FSMs, which is built on top of spatiotemporal video features and primitive object detection output. The Hidden Markov-Models and its variations are very popular approaches to activity recognition. One representative work is by Hongeng and Nevatia [3]. In their system, a semi-hidden Markov model is constructed by using the shape and motion features of tracked objects. Laxton et.al., [4] proposed a Dynamic Bayesian Network (DBN) structure which incorporates partially ordered sub-actions, a hierarchical action representation for building complex actions and an approximate Viterbi inference algorithm.

Since most graphical models are only assumed to handle sequential events, they are not capable of capturing activities with parallel actions. To address this problem, several contributions have been developed. Pinhanez [7] proposed one of the first works in modeling events with durations by incorporating Allen's interval algebra. Shi et.al., [9] proposed Propagation Networks (P-Nets), which model sequential activities with concurrent primitives. Different from many graphical models where primitives are considered instantaneous, primitives in these methods are assumed to have temporal durations such that interval logic can be applied.

In addition to the above mentioned graphical models, other state-based approaches have also been developed. Filipovyh and Ribeiro [2] presented a probabilistic model to capture human-object primitive interactions. In their framework, both static and dynamic appearances of the actor and the target object are encoded in a joint distribution. The intrinsic spatiotemporal configurations between actor and objects are also modeled.

Other approaches have also been pursued. Rao et.al. [8] presented a rank theory for matching trajectories. Activity trajectories are matched by analyzing the rank of their observation matrix. Boiman and Irani [1] have applied the spatiotemporal interest-point descriptor to detect irregular activities by comparing each descriptor with its neighbors in the spatiotemporal dimensions. Wong et.al., [12] have extended the probabilistic latent semantic analysis (pLSA) model to incorporate both visual parts and the structural information between visual parts to classify activities in videos. The major limitation here is that descriptor-based methods are not able to either capture the temporal order of the events or handle overlapping scenarios.

3. General Framework

As explained in the introduction, repetitive human actions are often observed in scenarios like retail checkout stations and factory assembly lines. Such human activity can be considered as a set of repeated sequential events (or *visual work units*), each of which is composed of a sequence of relatively isolated and separable primitives with strong spatiotemporal constraints. While a strict ordering is demanded between primitives, two consecutive *work units* may overlap to an arbitrary degree. This overlap comes as a natural byproduct of the rapid and repetitive nature of the activity.

In this section, we present a generative approach for grouping primitives into a set of repeated sequential events of interest. We are especially interested in addressing the issue of overlap. We first group the primitives into a large set of valid candidates for the sequential event of interest. By doing so, the overlap problem is resolved by considering the temporal orderings of the corresponding primitives

in consecutive sequential events. We then propose a Viterbi-like algorithm for selecting the most likely set of sequential events from the large pool of candidates for representing the data. In what follows, we assume that the primitives have already been obtained from some low-level event detectors.

3.1. Sequential Event Representation

We first give a general graphical representation for a sequential event. Let e_t^k be the k-th primitive in a sequential event that occurs at time t. A sequential event S is defined as a temporally ordered set of primitives $\{e_{t_1}^1, e_{t_2}^2, \ldots, e_{t_n}^n\}$ such that $t_1 < t_2 < \ldots < t_n$. For the purpose of clarity, we sometimes omit the superscript k in the text.

The sequential event is represented as a graph. As illustrated in Figure 2, a primitive e_{t_i} in a sequential event S is associated with an appearance node v_{t_i} that represents the visual information, and a location node l_{t_i} that denotes the spatial location of where the primitive occurs. The node C is a spatial model that places spatial constraints on the primitives. The primitives in a sequential event follow a Markovian model, such that the probability of S under the observation $O = (\mathbf{v}, \mathbf{l})$ is given by,

$$p(O|S) \propto p(\mathbf{v}|S)p(\mathbf{l}|S)$$
$$= p(v_{t_1}|e_{t_1}) \prod_2^n p(v_{t_i}|e_{t_i})p(e_{t_i}|e_{t_{i-1}}) \prod_1^n p(l_{t_i}|e_{t_i}) \quad (1)$$

where $\mathbf{v} = \{v_{t_1}, v_{t_2}, \ldots, v_{t_n}\}$ and $\mathbf{l} = \{l_{t_1}, l_{t_2}, \ldots, l_{t_n}\}$ represent the visual cues and spatial information respectively. Here, $p(v_{t_i}|e_{t_i})$ is the appearance likelihood model for the primitive e_{t_i} while $p(l_{t_i}|e_{t_i})$ is a spatial likelihood model for e_{t_i}. Term $p(e_{t_i}|e_{t_{i-1}})$ is the transition probability from primitive $e_{t_{i-1}}$ to primitive e_{t_i}.
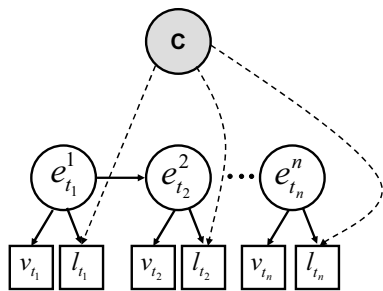

Figure 2. A graphical representation of the sequential event with n primitives. A primitive event node is associated with an appearance node that represents visual information, and a location node that indicates where the primitive occurs. The node C is a general model that places spatial constraints on the primitive nodes.

3.2. Building Sequential Events

Assume there are n sets of primitives $\{E_1, E_2, \ldots, E_n\}$ detected in a video sequence, where E_m is a set of primitives with a specific type m (e.g., all possible *Pickups* in the cashier scenario). Thus, a candidate sequential event S can be formed by selecting a primitive from each set with temporal order. We consider all such candidates by enumerating samples in $\{E_1, E_2, \ldots, E_n\}$. In particular, we represent all the candidates starting from a primitive $e_{t_i}^1 \in E_1$ by a tree rooted at $e_{t_i}^1$, which we call a *sequence tree* denoted by $Tr(i, :)$. In such a tree, any node at the j-th level is only selected from set E_j and all the children of the node occur in later primitive sets. This way, each path from the root of the tree to a leaf node corresponds to a candidate for a sequential event S. An example of such a sequence tree with 3 primitives is illustrated in Figure 3.

The number of sequential event candidates generated in this way grows exponentially with the number of primitives. To manage the size of the sequential event set in practice, heuristics could be applied to reduce the number of candidates dramatically. For instance, simple temporal constraints like requiring that two consecutive events occur within a specified time interval could prune out many impossible combinations.
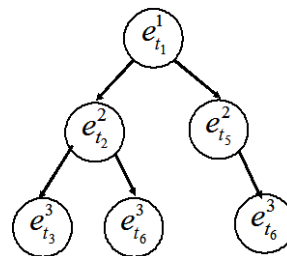

Figure 3. A sequence tree representing some potential sequential events with 3 primitives. $\{e_{t_1}^1, e_{t_2}^2, e_{t_3}^3, e_{t_4}^1, e_{t_5}^2, e_{t_6}^3\}$ is a set of primitive events detected in a video sequence. All candidates originate from primitive e_{t_1}.

3.3. Problem Formulation

The exhaustive combination scheme described above yields a great number of candidate sequential events. However, the majority of these are spurious, especially when the results of primitive detection are noisy. Our goal is to select a small set of sequential events that best match the truth in the data (see Figure 4(a)). We turn this selection process into an optimization problem where the strong temporal dependencies between primitive events and their spatial constraints are used to direct the optimization process.

A critical observation here is that although two sequential events may overlap, their corresponding primitives should not. We define two sequential events $S =$

$\{e_{t_1}^1, e_{t_2}^2, \cdots, e_{t_n}^m\}$ and $S' = \{e_{t'_1}^1, e_{t'_2}^2, \cdots, e_{t'_n}^m\}$ *disjoint*, or denoted as $S \cap S' = \emptyset$, iff

$$t_i < t'_i, \quad \forall i = 1 \ldots n, \tag{2}$$

Similarly, a set of sequential events $\mathbf{S} = \{S_1, S_2, \cdots, S_n\}$ is defined as *disjoint* if it satisifies:

$$\forall S_i, S_j \in \mathbf{S}, S_i \cap S_j = \emptyset, i \neq j \tag{3}$$

Given the above definitions, the genuine sequential events of interest in the video can be considered as a set of disjoint repetitive actions. Thus, in the context of Bayesian modeling, our task can be understood as identifying the most likely disjoint subsequence within some kind of model that best explains the observations emitted by the genuine set of sequential events in the data. This is an optimization problem and can be mathematically formulated as follows,

Let $\mathbf{S} = \{S_1, S_2, \cdots, S_n\}$ be the set of sequential event candidates ordered by time. Find a maximum disjoint subsequence $\hat{\mathbf{S}}$ within a model $\mathcal{M}(\theta)$ such that

$$\hat{\mathbf{S}} = \underset{\bar{\mathbf{S}} \in \mathcal{D}(\mathbf{S})}{\mathrm{argmax}} \; p(\bar{\mathbf{S}}|\bar{O}, \mathcal{M}(\theta)) \tag{4}$$

where $\mathcal{D}(\mathbf{S})$ is set of all possible disjoint subsequences of $\mathbf{S}$ and $\bar{O}$ is the corresponding observation of $\bar{S}$.

The optimization in Eqn.4 results in the maximized *throughput* of the target subject who invokes the events, which is encouraged in real-life scenarios (e.g., an employee who processes items fast tends to get rewarded).

Again, the repetitive sequential events are assumed to be a Markovian process. Based on the Bayes rule, we obtain,

$$
\begin{aligned}
p(\bar{\mathbf{S}}|\bar{O}, \mathcal{M}(\theta)) \quad &\propto p(\bar{O}|\bar{\mathbf{S}}, \mathcal{M}(\theta))p(\bar{\mathbf{S}}|\mathcal{M}(\theta)) \\
&= p(S_1)p(O_1|S_1)\prod_2^m p(O_i|S_i)p(S_i|S_{i-1})
\end{aligned}
\tag{5}
$$

where m is the length of the event series $\bar{\mathbf{S}}$. $p(O_i|S_i)$ can be further substituted by Eqn.1. $p(S_i|S_{i-1})$ is the transition probability between S_i and S_{i-1}.

There is much analogy between our problem and one of the problems considered by HMMs, which seeks an optimal state sequence to best explain the observations. However, a major difference here is that the optimization in our case is conducted over a subspace with only disjoint event sequences. This dramatically reduce the modeling complexity, making the optimization tractable. Also, the lengths of subsequences are varied while all state sequences have a fixed size in HMMs.

The selection of model $\mathcal{M}(\theta)$ depends on the specific problem under consideration. We will give a detailed example in Section 4.

3.4. Model Inference & Learning

The optimization problem described above seems intractable as the number of disjoint sequences grow exponentially with the number of sequential event candidates. However, it turns out that with some manipulation, a modified Viterbi algorithm can solve this problem efficiently.

We start by constructing a lower-triangular trellis with each sequential event being a node, as shown in Figure 4(b). The size of the trellis is $n \times l$ where n is the total number of sequential event candidates and l is the number of first primitives that correspond to the sequential event of interest. The construction embodies two important considerations, a) an event sequence is disjoint (*lower-triangular*) and b) the sequence is no longer than the number of first primitives detected *(n columns)*. In addition, *such a representation provides a path for any disjoint event sequences of interest.*
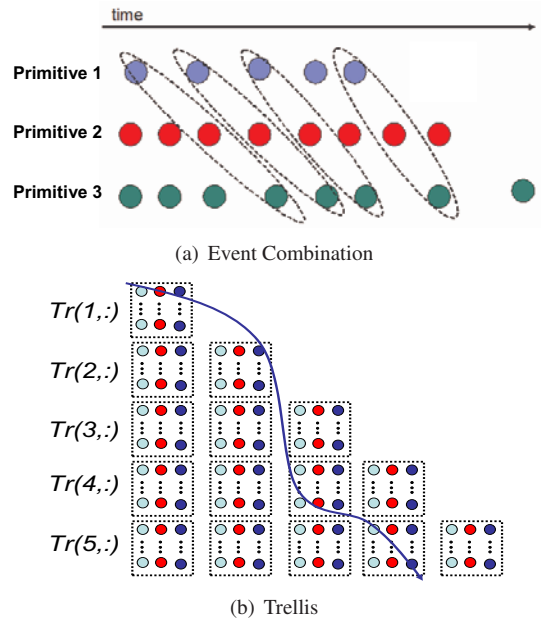

(a) Event Combination

(b) Trellis

Figure 4. (a) Given the primitive events detected, we are interested in identifying a set of disjoint sequential events of interest that correspond to the truth in the data. (b) A lower-triangular trellis can be formed by a set of sequence trees $\{Tr(1,:), Tr(2,:), \ldots, Tr(l,:)\}$ where $Tr(i,:)$ refers to the candidates starting at the ith sample in the set of the first primitives. The Viterbi algorithm can be used to find an optimal constrained path (blue line) for a maximum set of disjoint sequence with maximum likelihood.

Each node has an observation score computed from the model, and each two disjoint nodes in adjacent columns are associated with a transition probability by the model. A search for the optimal path is conducted in a similar way to a regular Viterbi algorithm, but only *constraint paths* with disjoint nodes are considered. Upon completion of the algorithm, each node is either isolated (no path to it), or set with the maximum likelihood resulting from an optimal sequence of events leading to that node. We locate the last

column with unisolated nodes, and start from the node with maximum likelihood in that column and backtrace the optimal sequence of sequential events. Since a path to a node in the jth column has a length of j, the path we've identified above is the most likely maximum disjoint subsequence that is pursued in Eq. 4.

Model learning is straightforward in our case as we can entirely rely on the HMM framework. The major difference is that we consider just part of the state space while a regular HMM considers the entire space.

4. Cashier Activity Modeling

Cashier activity recognition has significant applications in reducing retail shrink caused by employee-related fraud such as *sweethearting* mentioned in Section 1. In this section, we develop a spatio-temporal model based on the framework proposed in 3 for recognizing the predominant cashier activity in a retail context, which we term *visual scans*. As shown in Figure 1, this sequential activity includes 3 distinctive actions (or primitives), i.e, *pickup*, *scan* and *drop*, respectively. Our model integrates appearance information, temporal information and geometric cues into the framework discussed in Section 3.

4.1. Detecting Checkout-Related Primitives

We have developed an approach in [18] to detect the 3 primitive events of interest, i.e *pickup*, *scan* and *drop* in the checkout region. The approach is composed of two major parts: a motion-based event segmentation algorithm and an event recognition model based on Multiple-Instance Learning (MIL). The segmentation algorithm is used to identify segments in a video sequence as candidates for primitive events, which are further verified by the event recognition model. We briefly recap the fundamental ideas of the approach next.

The basic idea of the segmentation algorithm is to monitor the motion change (obtained by *frame differencing*) in a specific region of interest (ROI). Instead of thresholding the motion sequence directly, we take advantage of the temporal ordering of the primitives, which hints that there is one pickup (and drop) between two consecutive scans. We first use the motion peaks in the scan area as dividers to pre-segment the pickup (and drop) motion sequence. For each pre-segment between two scans, it is thresholded and the resulted sub-segments are further assessed with regard to duration, magnitude and motion patterns.

After segmenting the video sequence, we apply Space-Time Interest Points (STIP)to recognize events with the Bag of Features (BOF) model, similar to [16]. However, one problem arises when it comes to defining an appropriate ROI for the model due to the unoriented behaviors of the cashier who may pick up (or place) products anywhere in
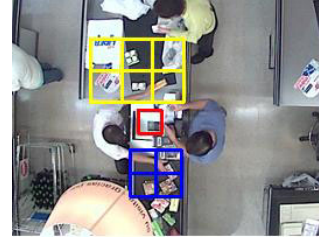

Figure 5. Multiple overlapped ROIs can address the location-sensitive issues appearing in the unoriented interactions between the cashier's hands and the transaction area. We used 4, 1 and 6 ROIs in the lead-in belt, scan and take-away belt areas respectively. For the purpose of illustration, the ROIs shown here are not overlapped.

the transaction area. Our idea is to place multiple regions to cover the entire transaction area so that each product is guaranteed to be placed in one region (Figure 5). The Multiple-Instance Learning (MIL) technique is adopted to resolve the unknown correspondence issue between primitives and their locations. More detail can be found in [18].

4.1.1 Temporal Models

Assuming that sequential events occur independently, the waiting time between two events can be modeled with the Erlang distribution,

$$f(t; k, \lambda) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!} \quad \text{for } t > 0 \qquad (6)$$

In our case, $k = 1$ as we are only interested in the time gaps between consecutive visual scans. So $p(S_i|S_{i-1})$ in Eq. 5 can be simplified as an exponential distribution,

$$p(S_i|S_{i-1}) = \lambda_s e^{-\lambda_s t} \qquad (7)$$

where t is the time gap between S_{i-1} and S_i.

The temporal dependencies between primitives, i.e, $p(e_{t_i}|e_{t_{i-1}})$ in Eq. 1 are also modeled by two separate exponential distributions λ_e^1 and λ_e^2. Note that the primitives of a visual scan follow each other closely, so λ^1_e and λ^2_e tend to be much larger than λ_s in general.

4.1.2 Geometric Models

Actions invoked by the cashier are limited by the reach of the cashier's arm(s). Thus, knowing where the cashier is during the occurrence of an event can help disambiguate spurious events that do not make geometric sense. We build a simple geometric model to capture the distance between the cashier and the location of an event.

Let l_{t_i} be the cashier's location when an event e_{t_i} is invoked in the kth ROI centered at r_k. Then the probability

of an event appearing at location P is written as,

$$p(l_{t_i}|e_{t_i}) \propto \mathcal{N}(x_{t_i}|\mu, \sigma) \qquad (8)$$

where $x_{t_i} = \overline{l_{t_i} r_k}$, i.e, the distance between the cashier and the location of the event. Note that the center of the ROI is only a rough approximation of the event location. We model pickup, scan and drop separately by using 3 Gaussian distributions under the assumption of independence.

The cashier's location is detected by background substraction. Basically, an ellipse is fitted to the contour of the largest blob obtained from background subtraction, and the center of the ellipse is considered as the cashier's location (See Figure 6).

(a) Likely cashier hand reach   (b) Unlikely cashier hand reach

Figure 6. The location of an event is limited within the reach of the cashier's arm. a) An likely reach and b) an unlikely reach. The ellipses show the locations of the cashier detected by background subtraction. The red boxes indicate where an event is invoked.

4.1.3 Learning Model Parameters

The set of parameters in the spatio-temporal model we have presented above is $\{\lambda_s, \lambda_e^1, \lambda_e^2, \{(\mu_i, \sigma_i)|i = 1 \ldots 3\}\}$. One big advantage of our approach over the others is that the learning and inference schemes of HMMs can be applied directly to our case with small modifications. The well-known Baum-Welch (i.e EM) algorithm used for learning HMMs gives the expected number of transitions n_{ij} from the ith state to j^{th} state, and the expected number π_i for the ith state. According to Section 3.4, n_{ij} and π_i are equivalent to the expected number of sequential event S_i followed by S_j, and the expected number of S_i observed, respectively. However, we only consider constrained paths in our case. Thus we update the temporal parameter λ_s in the M step by,

$$\lambda_s = \frac{1}{\sum_i \sum_j \delta_{ij} n_{ij} t_{ij} / \sum_i \sum_j \delta_{ij} n_{ij}} \qquad (9)$$

where

$$\delta_{ij} = \begin{cases} 0 & \text{if } S_i \cap S_j = 0 \\ 1 & \text{otherwise} \end{cases} \qquad (10)$$

t_{ij} is the time gap between S_i and S_j. λ_e^1 and λ_e^2 can be updated in a similar way.

We update the spatial parameters as follows,

$$\mu_j = \frac{\sum_i \pi_i^j x_i^j}{\sum_i \pi_i^j}$$
$$\sigma_j = \frac{\sum_i \pi^j (x_i^j - \mu_j)(x_i^j - \mu_j)}{\sum_i \pi_i^j} \quad j = 1 \ldots 3$$

$$(11)$$

where x_i^j is the distance between the cashier and the jth primitive event in the ith visual scan.

5. Experimental Results

5.1. Data & Evaluation Measures

We first evaluated our approach with a small data set (**ST-1**) including 10 videos captured from a real grocery store. The data involves 5 cashiers and each video corresponds to a single transaction. The number of items in the transactions varies from 6 to 29 with a mean of 10.7.

We further tested our approach by using a significantly larger data set (**ST-2**) recently captured from a lane in another grocery store. This data set includes transactions for an entire day with a total of 396 transactions and 5485 scanned items. The corresponding transaction logs were also recorded.

A large number of interventions on the part of the customer and bagger and dramatic differences in cashier behavior were observed in both data sets. There were also considerable occlusions in ST-2 caused by the head and body movement of the cashier.

For evaluation, we define the *overlap percentage* of a primitive e_1 and a prediction e_2 as their intersection divided by the union of the two events, i.e,

$$\tau = \frac{e_1 \cap e_2}{e_1 \cup e_2} \qquad (12)$$

A primitive event may relate to multiple predictions. We take the one with the maximum overlap percentage as the correct match if the percentage exceeds some threshold τ. All the others are considered as false positives. We counted the false positives and false negatives for each video, and computed the precision (p), recall (r) and F-measure ($f = 2*p*r/(p+r)$), accordingly. We set $\tau = 0.2$ for evaluating the results of the primitive detection.

For visual scans, a prediction is considered as correct only if all three of its primitives are successfully matched in the ground truth.

5.2. Evaluation Results on ST-1

We manually annotated the ground truth (start and end time) for each primitive in **ST-1**. The annotations were used to generate the ground truth for visual scans automatically by the combination algorithm discussed in Section 3.

We generated 10 data sets by permuting the 10 videos randomly. For each data set, 6 videos were used for training

Primitive	Alg.	Precision	Recall	F-measure
Pickup	ONE-R (70)	0.90 ± 0.06	0.91 ± 0.04	0.91 ± 0.03
	MIL-R(180)	0.84 ± 0.07	0.92 ± 0.03	0.88 ± 0.05
Scan	ONE-R (30)	0.91 ± 0.07	0.93 ± 0.05	0.92 ± 0.02
Drop	ONE-R (50)	0.81 ± 0.04	0.81 ± 0.05	0.81 ± 0.04
	MIL-R (160)	0.80 ± 0.03	0.87 ± 0.05	0.83 ± 0.02

Table 1. The best results (mean±variance) of using one ROI for each primitive (ONE-R) and multiple regions (MIL-R), achieved by the number of visual words indicated in the parenthesis.

Input	Alg.	Precision	Recall	F-measure
ONE-R	BASE	0.63 ± 0.07	0.54 ± 0.08	0.58 ± 0.08
	TEMP	0.78 ± 0.08	0.67 ± 0.08	0.72 ± 0.08
MIL-R	BASE	0.69 ± 0.08	0.63 ± 0.08	0.66 ± 0.08
	LOC	0.73 ± 0.06	0.68 ± 0.09	0.70 ± 0.08
	TEMP	0.83 ± 0.05	0.76 ± 0.07	0.80 ± 0.06
	LOC+TEMP	0.83 ± 0.06	0.77 ± 0.06	0.80 ± 0.06

Table 2. The recognition results of visual scans using visual (BASE), temporal (TEMP) information and geometric cues (LOC).

and the remaining 4 for testing. The results reported below were all averaged on the 10 data sets.

5.2.1 Primitive Detection

STIP features contain both edge information (**HOG**: Histogram of Oriented edges) and motion information (**HOF**: histogram of optical flow). Our previous experiments show that **HOF** outperforms **HOG** and the combination of both only leads to a slight performance improvement, largely due to the frequent and large background changes in the checkout area. Thus, we only reported the results based on **HOF**.

We placed 4, 1 and 6 ROIs in the customer lead-in belt, scan and the take-away belt areas respectively (See Figure 6). The ROIs are overlapped to cover the items placed on the boundaries of the ROIs. We didn't specifically tune the size of the ROIs, but ensured that they were large enough to capture the movement of the cashier's arm.

We compared the results of using one ROI for each primitive (ONE-R) and the MIL approach (MIL-R). In the case of applying one ROI only, we carefully tuned its position to avoid distractions from the bagging person as much as possible. As shown in Table 1, MIL-R demonstrates better recall than ONE-R, especially for the drop event in the take-away belt area where cashiers present more unoriented behaviors when placing products onto the belt. ONE-R yields higher precision, but this was achieved by particularly tuning the ROI to reduce the effect of the bagging activity.

5.2.2 Visual Scan Detection

We consider our combination algorithm using visual information only as the BASE algorithm, and incrementally add other information to the algorithm, including the location information of the cashier (LOC), temporal information (TEMP), and the combination of both (LOC+TEMP). The combination results of visual scan detection given in Table 2 were generated using the best primitive detection results reported in Table 1, from both ONE-R and MIL-R respectively. The geometric modeling is less meaningful in the case of a single ROI so no results were given on that. The temporal information significantly improves the disambiguating ability of the model and boosts performance by as much as 15%. The spatial information results
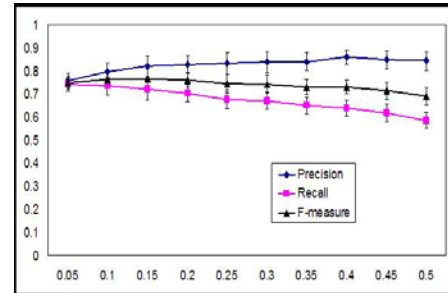

Figure 7. The Performance of primitive combination varies with the probability threshold τ for the primitives, which determines the number of primitives participating in the combination. A larger τ yields better precision and worse recall, and a lower F-measure overall. (Best viewed in color)

in a mediocre improvement. However, when combining it with temporal information, it does not yield better performance as expected. This is largely attributed to the cashier locator discussed in Section 4.1.2, which gave poor tracking of the cashier in a few videos due to distractions from the chair (which does not appear in the reference image) in the transaction area. In addition, the distinctive behavior differences between cashiers require a more sophisticated geometric model to capture the spatial constraints imposed by the cashier.

The most important parameter in our algorithm is the probability threshold τ set for the primitives, which determines the number of primitives considered in the combination. This threshold is not only directly related to the complexity of the algorithm. but also has direct impact on the performance. As shown in Figure 7, A larger τ yields better precision and worse recall, and a lower F-measure overall. The best results, shown in Table 2 were observed when $\tau = 0.1$.

5.3. Evaluation Results on ST-2

Due to the unaffordably high cost of detecting STIP features, it is impractical to take the event recognition approach discussed in Section 4 to evaluate a large data set like **ST-2**

(a) Recall Comparison
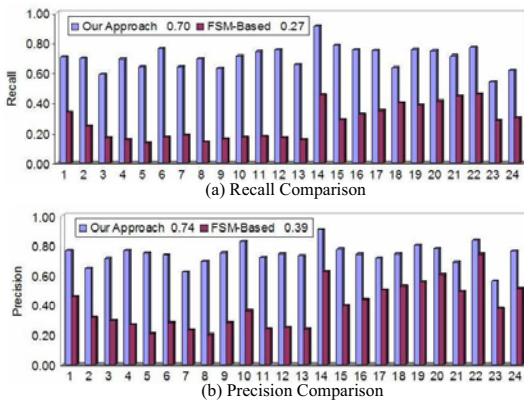

(b) Precision Comparison

Figure 8. Precision and recall achieved by our approach and a finite-state-machine model on the large data set. The results were averaged over all the transactions within half an hour.

(over 12-hour length of video). We therefore applied an efficient "push button" technique that has been implemented in our system to detect primitives in **ST-2**. This technique considers the pickup or drop event as a button-pushing-like action, recognizing these actions by identifying a focused beam of motion energy characteristic of the extension and retraction movement of cashiers arm with respect to a pre-specified region [13]. Due to limited space, we skip the details of this technique. A threshold-based motion detector was used to detect scan primitives.

The push-button method does not output the probability of an event, so we simply considered temporal constraints in our combination algorithm to detect visual scans in each transaction. The detected visual scans were further aligned to the transaction logs (TLOG) for evaluation in which a visual scan is considered legitimate only if a transaction record shows up during the scan. Figure 8 showed the recall and precision averaged over each half hour during the data (only busy periods are shown here). Our algorithm achieved over 70% accuracy in both precision and recall. Given the complexity of the data, the results are promising. As a comparison, we showed the results of a finite-state-machine model implemented in our system, which detects visual scans by considering temporal constraints locally. Our approach demonstrated clear advantage over the FSM model, which yielded less than half of the performance achieved by our algorithm.

6. Conclusions

We have presented a generative framework for detecting repetitive sequential events with strongtemporal dependencies and potential overlaps. We further demonstrated its effectiveness in a retail example where the predominant activity of the cashier has been analyzed. Our approach has

been evaluated on a large data set captured from a real grocery store. The results are encouraging given the various real-world complications present in the data.

Our current work focuses on improving the cashier tracking and developing a more sophisticated geometric model ([14, 15]) to capture the varied behavior differences between cashiers. In such a way, we expect to enhance the disambiguating ability of the spatio-temporal model we develop in this paper.

References

[1] O. Boiman and M. Irani, "Detecting Irregularities in Images and in Video", *ICCV*, 2005.

[2] R. Filipovych and E. Ribeiro, "Recognizing Primitive Interactions by Exploring Actor-Object States", *CVPR*, 2008.

[3] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden Markov models ", *ICCV*, 2003.

[4] B. Laxton, J. Lim and D. Creigman, "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video", *CVPR*, 2007.

[5] D. Mahajan, N. Kwatra, S. Jain and P. Kalra, "A Framework for Activity Recognition and Detection of Unusual Activities", *ICVGIP*, 2004.

[6] N. Oliver, E. Horvitz and A. Garg, "Layered representations for human activity recognition", *Int'l Conf. on Multimodal Interfaces*, 2002.

[7] C. Pinhanez, "Representation and Recognition of Action in Interactive Spaces", *Ph.D. Thesis*, 1999.

[8] C. Rao, A. Yilmaz and M. Shah, "View-Invariant Representation and Recognition of Actions", *IJCV*, Vol.50, Issue.2, 2002.

[9] Y. Shi, Y. Huang, D. Minnen, A. Bobick and I. Essa, "Propagation Networks for Recognizing Partially Ordered Sequential Actions ", *CVPR*, 2004.

[10] Toussaint, et.al., "Motion Segmentation Using Interface in Dynamic Bayesian Networks", *BMVC*, 2007.

[11] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey", *IEEE T-CSVT*, 2008.

[12] S-F Wong, T-K. Kim and R. Cipolla, "Learning Motion Categories using both Semantic and Structural Information", *CVPR*, 2007.

[13] R. Kjeldsen, "Polar Touch Detection", Interact, 2007 Beijing, China, July 2007.

[14] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition", IJCV, 2005

[15] D. Ramanan, D. Forsyth, A., Zisserman, "Tracking People by Learning their Appearance", PAMI, 2007.

[16] M. Schmid, C. Laptev, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR08*, 2008.

[17] S. Andrews, T. Hofmann, and I. Tsochantaridis, "Multiple instance learning with generalized support vector machines," *Artificial Intelligence*, 2002.

[18] Q. Fan, A. Yanagawa, R. Bobbitt, Y. Zhai, R. Kjeldsen, S. Pankanti, and A. Hampapur, "Detecting sweethearting in retail surveillance videos," in *ICASSP09*, 2009.