DETECTING SWEETHEARTING IN RETAIL SURVEILLANCE VIDEOS

Quanfu Fan, Akira Yanagawa, Russell Bobbitt, Yun Zhai, Rick Kjeldsen, Sharath Pankanti, Arun Hampapur

IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

ABSTRACT

A significant portion of retail shrink is attributed to employees and occurs around the point of sale (POS). In this paper, we target a major type of retail fraud in surveillance videos, known as *sweethearting* (or *fake scan*), where a cashier intentionally fails to enter one or more items into the transaction in an attempt to get free merchandise for the customer. We first develop a motion-based algorithm to identify video segments as candidates for primitive events at the POS. We then apply spatio-temporal features to recognize true primitive events from the candidates and prune those falsely alarmed. In particular, we learn location-aware event models by *Multiple-Instance Learning* to address the location-sensitive issues that appear in our problem. Finally, we validate the entire transaction by combining primitive events according to temporal ordering constraints. We demonstrate the effectiveness of our approach on data captured from a real grocery store.

Index Terms— retail shrink, event recognition

1 Introduction

Retail shrink is one of the topmost concerns on the minds of retailers. The shrink in stores is approximately 90B USD in the US and Europe alone. A significant portion of this shrink is attributed to retail fraud occurring around the Point of Sale (POS). While human surveillance has long been used to monitor transactions at the POS, it is not generally very effective and suffers from scalability issues. Data mining is another technique used to analyze transaction logs (TLOG) to infer cashiers' suspicious behaviors based on statistical analysis. However, these statistical anomalies may not be strongly correlated with fraudulent activity.

Recently video analytics have emerged as a promising technique for cashier fraud detection, thus becoming an effective approach for retail loss prevention [1, 2]. In this paper we focus on detecting one major type of retail fraud, known as *sweethearting* in the retail industry (or *fake scan*), using video analytics. *Sweethearting* occurs when a cashier purposely covers up the item barcode during the scan or passes the item around the scanner to avoid registering the item. As a result, the customer (usually a family member or friend of the cashier) is not charged for the *sweethearted* item. Sweethearting fraud is easy to commit but hard to catch, and is thus considered one of the most problematic types of fraud in retail sector.

To develop an effective approach for detecting sweethearting, we face a number of challenges. For example, the movement of the belt, bagging and customer interventions (Fig. 2), varied cashier behaviors, and low-resolution video capture, to name a few. Our approach first identify segments in a video sequence as candidates for primitive events at the POS by using a motion-based segmentation algorithm. The algorithm locates motion peaks in the scan region, which are used to distinguish events in the adjacent regions. The separated event segments are successively refined by thresholding, with temporal length, magnitude of motion and motion patterns taken into account. We then apply spatio-temporal features to recognize true primitive events from the candidates and prune those falsely alarmed. In particular, we learn location-aware event models by multiple-instance learning methodology to address the location-sensitive issues related to detecting events in a cluttered environment without generating excessive false positives. Finally, we validate the whole transaction process by combining primitive events according to temporal ordering constraints. The combination problem is formulated as an optimization problem and efficiently solved by a modified Viterbi algorithm. The final results are synchronized with the TLOG to flag fraudulent incidents in surveillance videos.

Definitions A typical process to transact one item at the POS includes three major actions from the cashier: picking up an item from the unload area, reading it via the scanner (or weighing an item if it has no bar code) and then placing the item on the exit area for bagging. Such a process is referred as a *visual scan* in this paper, and the 3 actions in order (i.e *pickup*, *scan* and *drop*) are the primary *primitive events* (or *primitives*) we consider.

Related work There has been a vast amount of research in event recognition in the computer vision literature, mostly related to human motion analysis [3]. Motion History Images (MHIs) ([4]) are an efficient temporal representation of human actions that capture both motion and shape. However it relies on good background subtraction. Recently spatio-temporal features [5, 6, 7] have drawn a lot of attention in event recognition and promising results have been reported in a number of applications such as [8, 9]. In addition, graphical models such as [10], CFG [11] and DBN [12] have been widely applied to model complex events by combining primitive events.

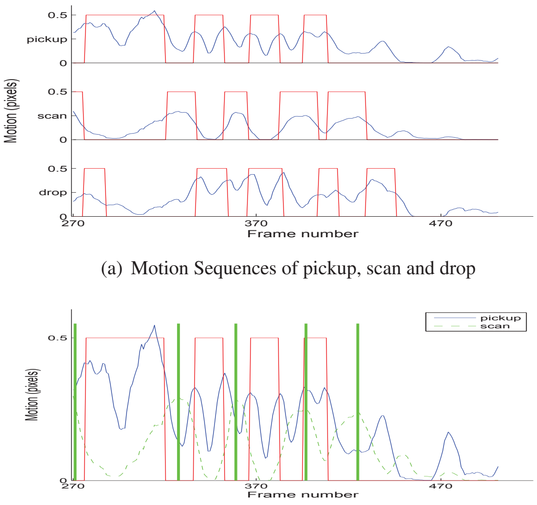
2 Segmentation of Video Sequences

To detect when events occur in a video sequence, one could apply a sliding window in time with a fixed scale (or multiple windows with varied scales) as done in [6]. However, this method is inefficient, and determining the scale (or scales) is non-trivial in many cases. We therefore develop an efficient algorithm to segment the video sequence and identify good candidates for primitives. The candidates can be further verified by more advanced event recognition algorithms.

The three primitives of interest can be simulated as an "in/out" process in which a hand or both hands enter and then exit a region quickly. We place a region of interest (ROI) for each primitive in the unload, scan and exit areas to capture this process. The motion pixels obtained by *frame differenc-*

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

ing are counted in each ROI for each frame and normalized by the area of the ROI. We observe some interesting patterns in the resulting motion sequences. For example, as illustrated in Fig. 1(a), most of the pickup (or drop) events display two peaks with a valley in-between, which faithfully depict the motion change caused by the interaction between the hand(s) and the specified region during an event. The valley corresponds to the moment of a short pause when the hand is about to reach an item (pickup) or to retrieve an item (drop). Note that the locations of the two peaks roughly correspond to the start and end time of an event. However, the valley is not always present if the hand moves too fast without pause, instead usually leading to a pattern of a single peak. Scan events show single peak patterns most of the time.



(a) Motion Sequences of pickup, scan and drop

(b) Event Dividers

Fig. 1. (a) Motion sequences of pickup, scan and drop (from top to bottom). The red boxes show the ground truth of the events. (b) Event dividers. The peaks identified in the scan motion sequence (green bold lines) effectively distinguish pickup (blue) events.

While the patterns indicated by the primitive events are visually identifiable, there is no easy way to segment them in the motion sequence. Fortunately, the temporal ordering of the events provides useful hints to help resolve this problem. Pickup, scan and drop occur sequentially, suggesting that there is one pickup (and drop) between two consecutive scans (Fig. 1(b)). Our algorithm thus first identifies scan events by thresholding the scan motion. The motion peak for each scan is located, and used as a divider to separate pickup and drop events. For each pre-segmented event, the algorithm further cuts off the motion sequence over a threshold, and assesses the resulting sub-segment(s) with regard to duration, magnitude and motion patterns. The details of the algorithm are skipped here due to limited space.

3 Recognition of Primitive Events

Space-Time Interest Points STIPs [5] are spatiotemporal features STIPs that are computed from local image points with both large intensity change and large variations in time. They roughly correspond to moments when there is abrupt motion change, such as stopping or starting. As shown in Fig. 2, several STIPs are detected near the cashier's hand at the moment when the hand is about to reach (pickup) or drop an item.

The STIPs detector automatically selects spatial and temporal scales with regard to the size and duration of the events. A spatio-temporal volume is formed for each STIP and further divided into grids of cuboids. Histograms of oriented gradient (HoG) and optic flow (HoF) are computed, normalized and concatenated into a local descriptor for each cuboid.

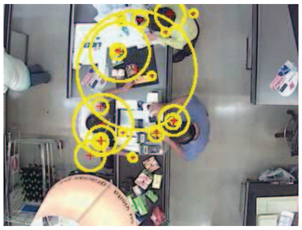


Figure 2. STIPs detected in one frame. Each STIP (circle) is associated with a location (center), spatial scale (radius) and temporal scale (not shown here). Several STIPs are detected near the hands of the cashier, but many are caused by the bagging person and the customer.

Bag-of-Features Model Similar to [9], our work uses Bag of Features (BOF) to represent events. To build a BOF model for an event, all the spatio-temporal features from a specified region (Fig.3(a)) are first clustered into k groups (*visual words*) based on their similarities. A histogram of the word occurrence frequency is constructed to form a compact representation of the event. The new histogram representation is used for classification with approaches such Support Vector machine.

Location-aware Event Modeling A drop event can be considered as an interaction between the cashier's hand(s) and the exit area. However, this interaction is unoriented, and can occur almost anywhere in the exit area. This poses a problem for defining an appropriate ROI for the event model. While an ideal ROI should be large enough to cover all possible locations of the events to be detected, it likely includes many irrelevant STIPs that result from the bagging person or the customer. To alleviate this problem, we apply the multiple-instance learning technique to build location-aware event models.

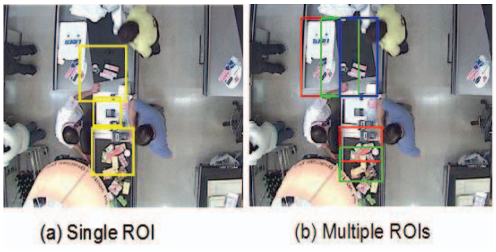


Fig. 3. An event model can be learned from either (a) a single ROI or (b) multiple overlapped ROIs.

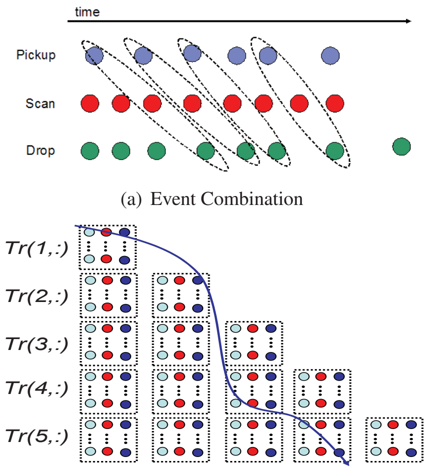
Our key idea is to use multiple overlapped ROIs to cover the transaction area as much as possible so that each event is guaranteed to be in one ROI (Fig.3(b)). However, the supervised learning paradigm discussed in Section 3 is not suited for multiple ROIs since the correspondence between events and ROIs is unknown. Instead, multiple-instance learning (MIL) has proven effective in resolving problems where correspondences are missing.

Multiple-Instance Learning (MIL) is proposed to solve the problem of learning from incompletely labeled data. Unlike supervised learning in which every training instance is associated with a label, MIL deals with data where labels (usually binary, either 0 or 1) are assigned to bags of instances instead of an individual instance. A *positive* bag has at least one positive instance that is related to a concept of interest while all instances in a *negative* bag are negative. The goal of MIL is to learn a model of the concept from the incompletely labeled data for classification of unseen bags or instances.

Learning event models from multiple ROIs is naturally connected to MIL in that each event corresponds to at least one ROI for sure, but the correspondence is not specified. For each annotated event, we can create a positive bag, the instances of which are the histograms of visual words from all the ROIs under the BOF representation. Negative bags can be generated in a similar way by considering those video segments with sufficient motion change but no primitives annotated in the ground truth. We use the SVM-based MIL algorithms (MIL-SVM) [13] to learn event models for pickup and drop. Scan events are more limited to a small region so we only use a single ROI for it.

4 Combining Primitive Events

Graphical models such as HMMs [10], CFGs [11], and DBNs [12] are commonly used for modeling complex events. In our case, any two primitive events, especially pickup and scan, may exist in parallel. It's not clear how this parallel structure can be captured by a graphical model. However, pickup, scan and drop occur in order, usually with short time gaps. In this section, we propose a novel approach for combining the primitive events into high-level events (i.e. visual scans) by considering their sequential ordering. In particular, we explore two types of temporal constraints: 1) time gaps between consecutive visual scans; and 2) duration of a visual scan.



(a) Event Combination

(b) Trellis

Fig. 4. (a) Given the primitive events detected, we are interested in identifying a set of disjoint triplets that correspond to the truth in the data. (b) A lower-triangular trellis formed by all possible triplets.

Let $P = \{P_1, P_2 \ldots, P_l\}$, $S = \{S_1, S_2 \ldots, S_m\}$ and $D = \{D_1, D_2 \ldots, D_n\}$ be the pickup , scan and drop events detected during a transaction, respectively. Also let $[t_s(E_i),\ t_e(E_i)]$ denote the start and end time of an event E_i. An event E_i is said to occur before another event E_j, i.e, $E_i < E_j$, *iff* $t_s(E_j) + \epsilon \geq t_e(E_i)$ where ϵ is a small non-negative number to tolerate detection errors.

We consider a triplet $Tr(i, j, k)$ as three primitives (P_i, S_j, D_k) that occur sequentially such that $P_i < S_j < D_k$ and $t_e(D_k) - t_s(P_i) \leq T$. T is a time threshold, which filters unlikely event candidates. We set $T = 10$ seconds. For convenience, we denote a group of triplets sharing primitives by replacing the corresponding indices by " : ". For example, $Tr(3, :, :)$ stands for all the triplets starting at P_3.

Two triplets $Tr(i_1, j_1, k_1)$ and $Tr(i_2, j_2, k_2)$ are disjoint, *iff* $P_{i_1} > P_{i_2}, S_{j_1} > S_{j_2}$ and $D_{k_1} > D_{k_2}$ or vice-versa. Note that this definition allows overlap of primitives in two consecutive triplets.

Given the above definitions, all the visual scans in a transaction can be considered as a sequence of disjoint triplets in temporal order(Fig.4(a)). Our goal is to identify such a set of triplets that is close to the true scan sequence as much as possible. Since time gaps between consecutive visual scans are short in general, we mathematically formulate the problem as follows,

Given 3 *sets of events (pickup, scan and drop), find a maximum set of* n *disjoint triplets in temporal order such that*

$$f(Tr_1, Tr_2, \ldots, Tr_n) = \sum_{i=2}^{n} d(Tr_{i-1}, Tr_i) \qquad (1)$$

is minimized.
Here $d(Tr_i, Tr_j)$ is the temporal distance between Tr_i and Tr_j given by

$$d(Tr_i, Tr_j) = (|t_s(P_j) - t_s(P_i)| + |t_e(D_j) - t_e(D_i)|)/2 \quad (2)$$

The optimization in Eqn. 1 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 will tend to get rewarded). The problem seems intractable as combinations of disjoint triplets grow exponentially with the number of primitives. However, it turns out, with some manipulation, a modified Viterbi algorithm [14] can solve this problem efficiently, in a similar spirit to HMM.

We generate a sequence of all triplets by concatenating $\{Tr(1, :, :), Tr(2, :, :), \ldots, Tr(l, :, :)\}$. We then construct a lower-triangular trellis with each triplet being a node, as shown in Fig.4(b). The trellis has a total of l columns. An edge is added between two triplets in adjacent columns if they are disjoint , and assigned a weight as the distance between the two triplets. Clearly, in such a representation there is a path for any set of disjoint triplets in temporal order. A Viterbi-like algorithm can then be applied to find an optimal path that minimizes Eq. 1. Our algorithm differs from the original Viterbi algorithm in that it considers only constrained paths between disjoint triplets. It can be proven that upon completion of the search, each node (or triplet) is either isolated (no path to it), or associated with an optimal path that entails a maximum set of disjoint triplets with minimum distance. Thus, we can start from the last unisolated node and backtrace to identify all the triplets corresponding to the visual scans in a transaction. Due to limited space, we skip the proof here.

Event	Alg.	Precision	Recall	F-measure
Pickup	**SEG**	0.42	0.96	0.58
	BOF	0.84 ± 0.09	0.90 ± 0.04	0.86 ± 0.05
	MIL-BOF	0.87 ± 0.11	0.88 ± 0.04	**0.87** ± 0.06
Scan	**SEG**	0.70	0.99	0.82
	BOF	0.88 ± 0.06	0.96 ± 0.03	**0.92** ± 0.03
	MIL-BOF	-	-	-
Drop	**SEG**	0.56	0.94	0.71
	BOF	0.76 ± 0.09	0.90 ± 0.06	0.82 ± 0.07
	MIL-BOF	0.81 ± 0.06	0.91 ± 0.06	**0.86** ± 0.05
Visual Scan	**COMB(SEG)**	0.65	0.93	0.76
	COMB(BOF)	0.88 ± 0.05	0.82 ± 0.03	0.84 ± 0.02
	COMB(MIL-BOF)	0.92 ± 0.06	0.81 ± 0.05	**0.86** ± 0.05

Table 1. Results of primitive event detection from **SEG**, **BOF** and **MIL-BOF**(using HOF and HOG features). The scan area is relatively small so only a single ROI was used.

5 Experiments And Results

Data We experimented with 10 videos captured from a real grocery store. The data involve 5 cashiers and each video corresponds to one transaction. The number of items in the transactions varies from 6 to 29 with a mean of 10.7.

We manually annotated the ground truth (start and end time) for each primitive. The annotations were used to generate the ground truth for visual scans automatically by the combination algorithm discussed in Section 4.

We generated 10 data sets by permuting the 10 videos randomly. For each data set, 6 videos were used for training and the remaining 4 for testing. The results reported below were all averaged on the 10 data sets.

Event Detection For evaluation, we define the *overlap percentage* of a primitive event and a prediction as their intersection divided by the duration of the primitive itself. 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 $\tau = 0.25$ for evaluating the results of the primitives, and $\tau = 0.5$ for the combined visual scans.

We considered 3 algorithms for primitive event detection: the segmentation algorithm (**SEG**), the BOF model with one single ROI (**BOF**) and the BOF model with multiple ROIs (**MIL-BOF**). We used 3 ROIs for drop events in the exit area, and 2 ROIs for pickup events in the unload area (Fig.3(b)). The ROIs were placed in a way to capture the dominant motion direction for a given event. We also presented the results of the combination algorithm. **COMB**(x) denotes the combination algorithm using input from primitive detector x.

Due to the frequent and large change of background in the transaction area, we found that HOG features contribute little to the performance, hence we only reported the results of different algorithms using HOF features only. As shown in Table 1, the segmentation algorithm (**SEG**) achieves high recall, which offers good input to the spatio-temporal models (**BOF** and **MIL-BOF**). **MIL-BOF** produces better results than **BOF**, especially for the drop events that are notably affected by the bagging person and customers. When combining all 3 primitive events (using the best results produced by each detector), **MIL-BOF** performs the best.

Sweethearting Detection We also tested our algorithm on another data set (8 transactions) captured from a grocery store. The TLOG is available for this data set, including 209 scanned items 29 random fake scans staged by 2 cashiers. Due to limited space, we only reported the results of sweethearting detection in Table 2, which were generated by matching the combined visual scans to the TLOG. The combination algorithm demonstrates promising capability at catching sweethearting, though yielding higher false positive rate.

Alg.	Precision	Recall	F-measure
COMB(MIL-BOF)	0.42 ±0.10	0.88 ±0.12	0.56 ±0.12

Table 2. Performance of *sweethearting* detection using the visual scans resulted from **MIL-BOF**.

6 Conclusions

We present an approach based on spatial-temporal features to detect sweethearting in surveillance videos. Our approach demonstrates good performance in recognizing checkout-related events at the POS in the presence of various complications from the real world. We also propose an approach to combine sequential primitive events into high-level events. Currently we are extending our approach to a probabilistic model to improve its disambiguating ability.

7 Acknowledgements

We would like to thank Lexing Xie and Rogerios Feris for helpful discussions on event modeling.

8 References

[1] StopLift, “http://www.stoplift.com/,” .

[2] intellivid, “http://www.intellivid.com/,” .

[3] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” in *IEEE Transactions on Circuits and Systems for Video Technology*, 2008, pp. 1473–1488.

[4] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE PAMI*, vol. 23, pp. 257–267, 2001.

[5] I. Laptev and T. Lindeberg, “Space-time interest points,” in *ICCV*, 2003, pp. 432–439.

[6] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features,” in *ICCV05*, 2005.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005.*, 2005, pp. 65–72.

[8] R. Filipovych and E. Ribeiro, “Recognizing primitive interactions by exploring actor-object states,” in *CVPR08*, 2008.

[9] Marszalek M. Schmid C. Laptev, I. and B. Rozenfeld, “Learning realistic human actions from movies,” in *CVPR08*, 2008.

[10] H. H. Phung D.Q. Venkatesh S. Duong, T.V.; Bui, “Activity recognition and abnormality detection with the switching hidden semi-markov model,” in *CVPR05*, 2005, pp. 1709–1718.

[11] M. S. Ryoo and J. K. Aggarwal, “Recognition of composite human activities through context-free grammar based representation,” in *CVPR*, 2006, pp. 1709–1718.

[12] B. Laxton, J. Lim, and D. Kriegman, “Leveraging temporal, contextual and ordering constraints for recognizing complex activities,” in *CVPR07*, 2007.

[13] S. Andrews, T. Hofmann, and I. Tsochantaridis, “Multiple instance learning with generalized support vector machines,” *Artificial Intelligence*, pp. 943–944, 2002.

[14] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” in *IEEE Transactions on Information Theory*, 1967, pp. 260–269.