DETECTING HUMAN ACTIVITIES IN RETAIL SURVEILLANCE USING HIERARCHICAL FINITE STATE MACHINE

Hoang Trinh, Quanfu Fan, Jiyan Pan, Prasad Gabbur, Sachiko Miyazawa, Sharath Pankanti

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

ABSTRACT

Cashiers in retail stores usually exhibit certain repetitive and periodic activities when processing items. Detecting such activities plays a key role in most retail fraud detection systems. In this paper, we propose a highly efficient, effective and robust vision technique to detect checkout-related primitive activities, based on a hierarchical finite state machine (FSM). Our deterministic approach uses visual features and prior spatial constraints on the hand motion to capture particular motion patterns performed in primitive activities. We also apply our approach to the problem of retail fraud detection. Experimental results on a large set of video data captured from retail stores show that our approach, while much simpler and faster, achieves significantly better results than state-of-the-art machine learning-based techniques both in detecting checkout-related activities and in detecting checkout-related fraudulent incidents.

Index Terms— video signal processing, finite state machine, retail shrink, event recognition

1. INTRODUCTION

In retail, a significant loss of over billions worldwide each year is mediated by employees and customers due to checkout-related fraud (or occasionally operational errors) around the point of sale (POS). Video analytics systems have recently been considered a very effective tool for detecting retail fraud, gaining more and more preference over human surveillance, due to their advantage in efficiency and scalability [1, 2, 3].

To register an item at the POS, the cashier usually performs three activities: picking the item from the lead-in belt (*pickup*), passing the item through scanning devices such as barcode reader, weighing scale, etc (*scan*), and placing the item on the exit/bagging area (*drop*). Such a process is usually referred to as a *visual scan*, and the three aforementioned activities are considered *primitives*. A key component of retail fraud detection systems is the detection of such primitive activities, which still remains a challenging problem. Emerging approaches to for human activity recognition using spatio-temporal features [4, 5, 6] or motion history [7] in combination with learning algorithms can be applied. However, these approaches are computationally expensive and sensitive to noisy and low resolution video data. Moreover, the system may have to be retrained for different types of lane layout, different camera viewing angles, and/or even for different cashiers. These disadvantages make them unsuitable for a large-scale deployment of retail surveillance applications. Other approaches have turned to more efficient low-level video processing techniques, based on extracting foreground mask using background subtraction [8] or motion map using frame differencing [9]. However these approaches suffer from motion noise caused by different sources. In the retail example, these can be checkout belt movement, lighting change and customer interactions.

In this paper, we propose a highly efficient and accurate vision technique to detect checkout-related primitive activities. We observe that the primitives *pickup*, *scan*, *drop* follow a specific motion pattern - each primitive is an in/out process in which the hand enters and exits a region of interest (ROI). Based on this observation, our approach aims at detecting activities following such a motion pattern. To detect hand motion, first, a hand color model is adaptively learned from continuously collected hand pixel examples. Based on this model, a hand motion map is formed by classifying each motion pixel obtained by frame differencing as hand pixel or not. This map explicitly captures the hand motion only and eliminates a lot of motion noise from belt movement, background changes, customer interactions, etc. We then design a hierarchical FSM [10] to check whether or not the hand motion follows the in/out pattern. The first-level FSM combines the extracted hand motion with prior spatial constraints to generate a sequence of hand motion states. The second-level FSM uses this state sequence as input to verify the in/out hand motion pattern. We apply our algorithm to detect all 3 types of primitives in retail transactions. Our approach can work well in very different types of lane layouts, and requires little tuning effort. It is also robust to camera view changes, image noise and motion noise.

We evaluate our approach on a large set of real cashier checkout activities captured from retail stores. Experiments show that our approach significantly outperforms more sophisticated state-of-the-art machine learning-based techniques in primitive event detection, while being equally efficient as other real-time techniques ([9]) and requiring much less tuning efforts. The results also demonstrate that the

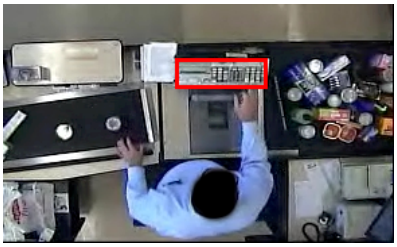


Fig. 1. An image region used for collecting new hand pixel samples. (*Best viewed in color*)

improvement in detecting primitives leads to the significant improvement in detecting cashier-related fraudulent incidents at checkout counters.

2. OUR APPROACH FOR DETECTING PRIMITIVE ACTIVITIES

2.1. Hand Motion Extraction

To estimate hand motion, a crucial step is to detect and locate the hand, which remains a challenging task in many real-life scenarios. With real-time and large-scale video surveillance systems, it is quite inefficient to apply sophisticated hand detection and tracking methods [11, 12]. We decide to use a relatively simple and efficient approach for hand motion detection, based on a hand color model and motion cues, similar to [13]. For the hand color model, whatever initial color model we use would gradually become inaccurate due to illumination changes, or suddenly become invalid due to cashier switching. Therefore we develop a solution to learn the hand color model adaptively. We define a particular region where the hand usually enters during a transaction (e.g. keyboard, touchscreen) (Figure 1). A motion map is computed by thresholding the result of frame differencing. By capturing a snapshot when the motion map overlaps with this region, we can continuously collect new hand pixel samples. Each sample is a rectangle image patch around the center point of the overlapping area. After a sufficient number of new samples are collected, the new hand color model is estimated as follows:

- Perform K-means clustering in RGB space to partition all pixels in all examples into k clusters.
- Select the cluster with maximum size.
- Compute mean μ_h and the 3×3 covariance matrix Σ_h for all pixels in the selected cluster. Our new hand color model is the pair of (μ_h, Σ_h).

Next, a hand color classifier is applied to all the image pixels captured in the motion map to label each pixel as hand pixel or not. The classifier uses (μ_h, Σ_h) to compute a hand likelihood value for each pixel. Then a threshold is applied to convert the likelihood value to a binary pixel label. We then apply morphological operators to this binary map to eliminate more noise. Figure 2 illustrates some detected hand motion maps.

Fig. 2. Hand motion maps detected by our approach. Blue pixels are labeled as hand pixels by the classifier. (*Best viewed in color*)

2.2. Hierarchical FSM for Motion Pattern Recognition

The intuition of our approach is illustrated in Figure 3. If we put a sequence of parallel lines at the border of the pickup area, when entering/exiting the pickup area, the hand has to cross all these lines one by one in a sequential order. Therefore, in order to detect a pickup primitive activity, which is composed of an entering followed by an exiting hand motion w.r.t the pickup area, we need to determine such interactions between the hand and the line sequence.

We define a deterministic FSM $F_i^{(1)}$ for each line i in a sequence (Figure 4(a)) to keep track of the state of the line. $F^{(1)}$ is defined by a quadruple $(\Sigma_1, S_1, s_1^0, \delta_1)$ where:

- $\Sigma_1 = \{0, 1\}$ is the input alphabet. 1 indicates the line is contacted by the hand (the line is on), 0 indicates otherwise (off).
- $S_1 = \{(00), (01), (10), (11)\}$ is the set of states. (01) indicates that the line is switched on, i.e. changes its state from 0 to 1, (10) indicates it is switched off, (00) and 11 indicate the line stays at its current state.
- $s_1^0 = (00)$ is the initial state.
- $\delta_1 : S_1 \times \Sigma_1 \rightarrow S_1$ is the state-transition function visually described in Figure 4(a).

We also define another deterministic FSM $F^{(2)}$ to determine the interaction of the hand w.r.t the line sequence. $F^{(2)}$ is defined by a quadruple $(\Sigma_2, S_2, s_2^0, \delta_2)$ where:

- $\Sigma_2 = \{s_1, ..., s_n, -s_1, ..., -s_n\}$ is the input alphabet. s_1 indicates line 1 in the sequence is switched on, $-s_1$ indicates it is switched off. n denotes the number of line in a sequence.
- $S_2 = \{I_1, ..., I_{n-1}, IN, O_n, ..., O_2, OUT\}$ is the set of states. I_1 indicates that line 1 is switched on, i.e.

(a) (b)

Fig. 3. (a) An example of line configuration: base lines are put at borders of ROIs to impose spatial constraints on the hand motion. (b) Line sequences automatically generated from the base lines on the left. (*Best viewed in color*)

at state (01). O_2 indicates line 2 is switched off, i.e. at state (10). IN indicates an entering hand motion is accepted. OUT indicates an exiting hand motion is accepted.

- $s_2^0 = OUT$ is the initial state.
- $\delta_2 : S_2 \times \Sigma_2 \rightarrow S_2$ is the state-transition function visually described in Figure 4(b).

A primitive activity is accepted when $F^{(2)}$ completes a loop through all the states and returns back to the initial state OUT. As illustrated in Figure 4(b), the input alphabet of $F^{(2)}$ is determined by a set of $F^{(1)}$ machines. Therefore our model can be considered a hierarchical FSM, in which the first-level FSMs produce input for the second-level FSM.

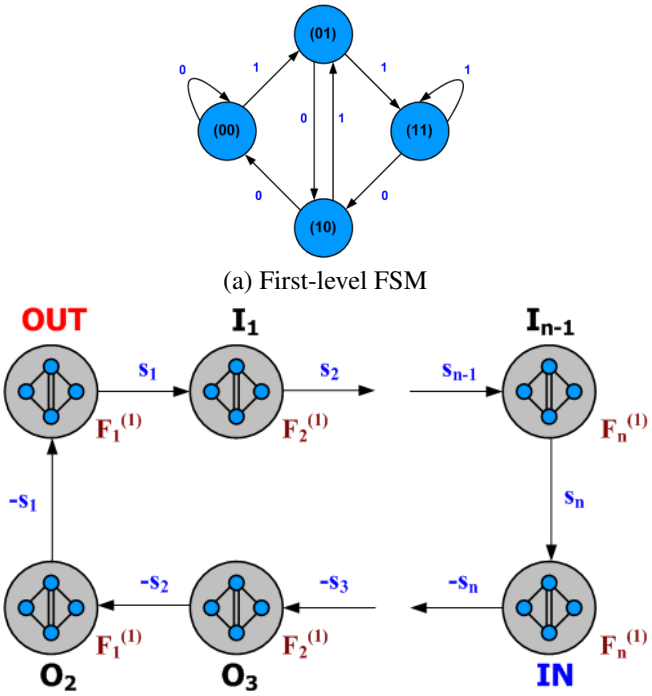
2.3. Line Sequence Configuration

For each sequence, we only need to define one base line (The red lines in Figure 3(a)). Intuitively these base lines should be located at the border of the ROIs. Starting from each base line, a line sequence is automatically generated (Figure 3(b)), based on some predefined parameters such as the number of lines n_l, and the distance between each line d_l. The overall performance of our method also slightly depends on these parameters. For example, the distance between the lines (in pixels) should be proportional to the image resolution. Specifically in our experiments, we empirically set $n_l = 3$ and $d_l = 10$.

3. EXPERIMENTAL RESULTS

3.1. Primitives and Visual Scan Detection

We evaluate our approach on the same dataset used in [3] - a set of video data of real checkout activities with three different cashiers captured from retail stores. The videos are at high frame rate (20 FPS) and low resolution (320x240), each video corresponds to one transaction. Our result is directly



(a) First-level FSM

(b) Second-level FSM

Fig. 4. (a) First-level FSM determines the states of each line in a line sequence. (b) Second-level FSM uses the output states from the first-level FSM to determine the in/out state of the hand motion.

comparable with the results from two different methods reported in [3]: the Bag of Features model (BOF) with one single ROI, and the BOF model with multiple ROIs, using multiple instance learning (MIL-BOF). Due to the small scan area, only BOF was applied for scan. More details can be found in [3]. The results reported in Table 1 show that our efficient approach outperforms more sophisticated machine learning-based techniques in detecting *pickup* and *drop* primitives. For *scan*, our approach produces slightly more false positives than the BOF method (lower precision), probably because sometimes the cashier has to scan an item multiple times before the scan is successful. Finally, after the primitives are detected, we use the Viterbi-like algorithm proposed in [2] to combine each triplet of *pickup*, *scan* and *drop* into a *visual scan*. Our approach achieves the best precision, recall and F-measure in detecting *visual scan*.

3.2. Checkout-related Fraud Detection

We use the Hungarian algorithm to match the list of detected *visual scans* with a list of barcode signals received from the transaction log (TLOG). A *visual scan* is validated if it is matched with a barcode signal. Otherwise it is an invalid *visual scan* which corresponds to a checkout-related fraud.

We tested our algorithm on a very large set of real check-

Activity	Alg.	Precision	Recall	F-measure
Pickup	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
	Our method	**0.84**	**0.96**	**0.90**
Scan	BOF	0.88±0.06	0.96±0.03	0.92±0.03
	MIL-BOF			
	Our method	**0.83**	**0.96**	**0.89**
Drop	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
	Our method	**0.92**	**0.86**	**0.89**
Visual Scan	BOF	0.88±0.05	0.82±0.03	0.84±0.02
	MIL-BOF	0.92±0.06	0.81±0.05	0.86±0.05
	Our method	**0.95**	**0.82**	**0.88**

Table 1. Comparison between our performance and the methods in [3] in terms of primitives and *visual scan* detection.

Alg.	False Positive Rate	Recognition Rate
[2]	3.8%	46%
Our method	**2.5%**	**62%**

Table 2. Comparison between our performance and the method in [2] on a very large dataset in terms of checkout-related fraud detection.

out activities captured from two retail stores for one business day, covering a variety of lane types and settings, with multiple different cashiers. For this dataset, we obtained ground truth by manually annotating all true cases of checkout-related fraud in all the videos. Unfortunately, many conceptually true invalid *visual scans* such as cashier passing by bags or baskets, scanning items with late or early barcode, were not annotated as true cases in the ground truth, due to limited annotation resources. Therefore such cases will be considered false positives if detected. We compare our results with the state-of-the-art system in [2]. The results reported in Table 2 shows that or approach produces significantly better results in terms of both false positive rate and recognition rate.

3.3. Scalability Analysis

A scalability test was conducted with our video analytics system on a PC Quad Core 2.4Ghz, 3.0 GB RAM, Windows XP, demonstrating that our system can monitor in real time 20 live video streams at 15 FPS and (360x244) resolution. Another test on an enterprise server in a real store with our system running shows that our system can monitor on average 2 live cameras per CPU, at the same video frame rate and resolution.

4. CONCLUSION

In this paper, we introduced a very efficient, accurate and robust algorithm to detect checkout primitive activities. Experimental results show that our new approach, although simpler and more intuitive, outperforms more sophisticated machine learning-based techniques both in detecting primitives and more complex checkout activities. In the near future we will focus on extending the approach to detect more cashier-customer interactions.

5. REFERENCES

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

[2] Q. Fan; R. Bobbit; Y. Zhai; A. Yanagawa; S. Pankanti and A. Hampapur, "Recognition of repetitive sequential human activity," in *CVPR*, 2009.

[3] Q. Fan; A. Yanagawa; R. Bobbit; Y. Zhai; R. Kjeldsen; S. Pankanti and A. Hampapur, "Detecting sweethearting in retail surveillance videos," in *ICASSP*, 2009.

[4] I. Laptev and T. Lindeberg, "Space-time interest points," in *ICCV*, 2003.

[5] I. Laptev; M. Marszalek; C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[6] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *CVPR*, 2010.

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

[8] F. Nater; H. Grabner and L. Van Gool, "Exploiting simple hierarchies for unsupervised human behavior analysis," in *CVPR*, 2010.

[9] Q. Fan; A. Yanagawa; R. Bobbitt; Y. Zhai; S. Pankanti and A. Hampapur, "Fast detection of retail fraud by using polar touch buttons," in *ICME*, 2009.

[10] Tom M Mitchell, *Machine Learning*, WCB/McGraw-Hill Corporation.

[11] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *IJCV*, vol. 29, pp. 5–28, 1998.

[12] E. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *International Conference on Automatic Face and Gesture Recognition*, 2004.

[13] J. Alon; V. Athitsos; Q. Yuan and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE PAMI*, vol. 31, pp. 1685–1699, 2009.