IDIAP RESEARCH REPORT

Detecting queues at vending machines: a statistical layered approach

Xavier Naturel [a] Jean-Marc Odobez [a]

IDIAP–RR 08-04

April 2008

[a] IDIAP Research Institute, P.O. Box 592, Centre du Parc 1920 Martigny, Switzerland.

IDIAP Research Institute **www.idiap.ch**
Av. des Prés–Beudin 20 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 08-04

Detecting queues at vending machines: a statistical layered approach

Xavier Naturel Jean-Marc Odobez

April 2008

Abstract. In this report, a method for monitoring activity at a ticket machine is presented. While this work has been done in the specific context of Turin metro, the proposed model could be applied to other locations and tasks in video-surveillance. Monitoring the activity is based here on event recognition, by modelling directly the events of interest.
We especially focus on detecting queues at ticket vending machines. A 2-layer model is proposed. In the first layer, several sub-events are defined and detected using a discriminative approach (SVMs). The second layer uses the result of the first and model the high-level event of interest. Results are assessed on 4 hours of real video footage coming from Turin metro station.

1 Introduction

Our overall objective is to monitor the general usage of a metro station equipment, with an emphasis on ticket vending machine (machine usage, machine mis-use or vandalism, anomalies, etc), and extract their statistics. It is hoped that automatic generation of statistics of the station's usage will provide a better understanding for the end-users. We focus on the recognition of one specific event (queues), to illustrate our approach, but some other events are also detected along the way, which can provide interesting insights.

One motivation of this work is to monitor a metro station by detecting events, by modelling them directly. One interest of this method is to avoid complex methods like tracking, which are computationnaly intensive and can also have difficulties with even medium crowding situation. The proposed approach is a layered model, where a first layer is used to model some sub-events related to a higher level event, which is modelled by the second layer. This layered approach is proposed here in a context of detecting queues at a ticket machine, but the framework could be applied to another task. In previous approaches [6, 2], the 2 layers were using HMM. It has been chosen in this work to use SVM classifiers for the first layer. One of the interest is that SVMs can work with high-dimensionnal features, and do not require as much training data as HMMs.

The set-up we are using for our experiments is illustrated in Figure 1. Two cameras are looking at the Turin DOD station hall, where two vending machines are visible.

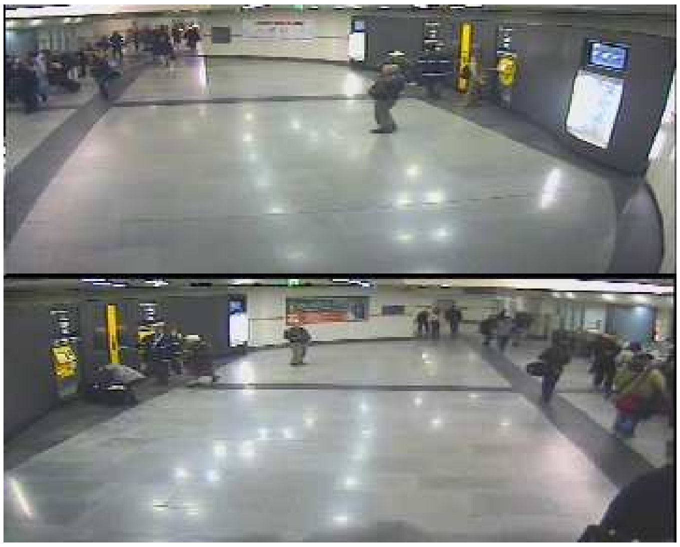

Figure 1: Camera views in Turin metro station, with the 2 ticket machines to be monitored.

2 Task definition and overall approach

A queuing event is objectively defined as people waiting to access the vending machine. This definition implies the following elements. First, waiting people *must* use the vending machine after it has become available. Otherwise, if the machine is free and they do not access it, we do not consider that there is a queuing event. Secondly, in our data, there are unfortunately few instances of several people waiting and actually forming a physical queue that can be identified from a single image (something which might be difficult as well given the camera viewing angle). In practice, this means that the event is essentially defined by temporal relationships between sub-events: some people are waiting while the machine is occupied; then people who were using the machine are leaving, and people are entering

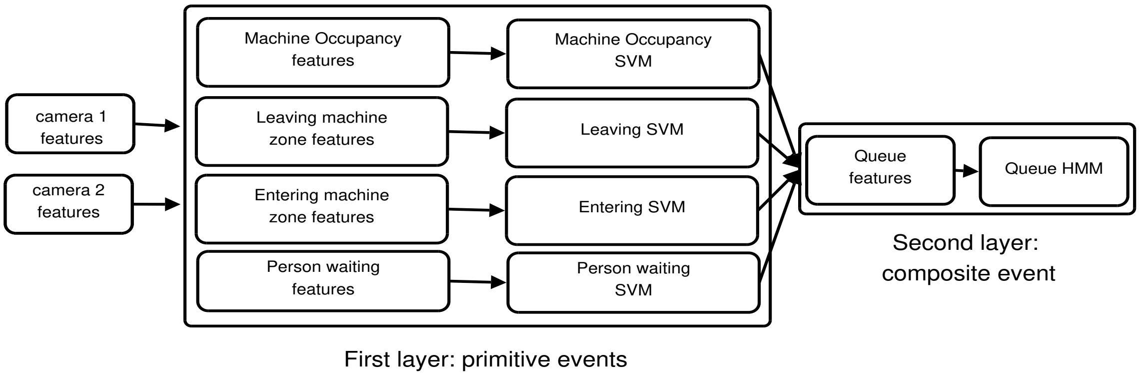

Figure 2: The layered approach.

(i.e. approaching) the machine zone to use it, while there still are people waiting. This sequence of events can continue until no more people are waiting. As a consequence, queuing can be defined by a set of 4 sub-events (usually called primitive events in the ontology literature), that our system will recognize. These are: { *machine occupancy (people in zone* $\mathbf{z}_m$*), entering the zone* $\mathbf{z}_m \cup \mathbf{z}_w$*, leaving the zone* $\mathbf{z}_m \cup \mathbf{z}_w$*, people waiting in the waiting zone* $\mathbf{z}_w$ }, where the machine zone $\mathbf{z}_m$ denotes a region just in front of the vending machine, while $\mathbf{z}_w$ corresponds to a further region of the hall (see figure 3).

To recognize a queuing event, we propose to use a layered approach [6, 2] based on machine learning techniques. More precisely, in a first layer, separate modules will detect the primitive events, and the classification results of these primitive events will be fed into a higher level model, corresponding to the second layer. In [6, 2], the first layer is built using HMM models. However, in our case, the feature set is quite large and we do not have a large number of positive example for training data. Using a generative model like HMM to do recognition might not be appropriate. It is proposed here to rely on a discriminative approach instead. More specifically, the first layer is composed of Support Vector Machines (SVM) classifiers, which take raw features as input, and output classification scores for each of the defined 4 sub-events (Machine occupancy, Leaving zone, Entering zone, and waiting in zone). These scores are then used as input features to the higher level model, here an Hidden Markov Model (HMM). This layered approach is illustrated by figure 2.

3 Features definition and recognition model for primitive events

3.1 Feature definition

To compute the features, two sets of 3D cylinders are defined on the ground plane, one for each ticket machine, roughly corresponding to an average human height and width (1m80 high and 80 cm wide). These 3D cylinders coordinates are then projected into the image plane of each camera, using the projection matrix obtained through calibration. This results in a set of 2D bounding boxes as illustrated on figure 3.

For each vending machine, the set of boxes is composed of a first row of 3 boxes in zone $\mathbf{z}_m$, and a regular grid of 3 by 5 boxes behind this front row, in the waiting zone $\mathbf{z}_w$. Figure 3 shows the boxes setup for one ticket machine, on an image from one camera view, as well as a schematic view of the boxes position, and the machine and waiting zones $\mathbf{z}_m$ and $\mathbf{z}_w$. Similar zones and bounding boxes are defined for the second ticket machine.

Background subtraction using the technique from [5] is performed on each camera view, and the resulting images are binarized. This results in an image with pixels of value 1 for foreground and 0 for background. For each box, the percentage of foreground pixels in this particular bounding box is

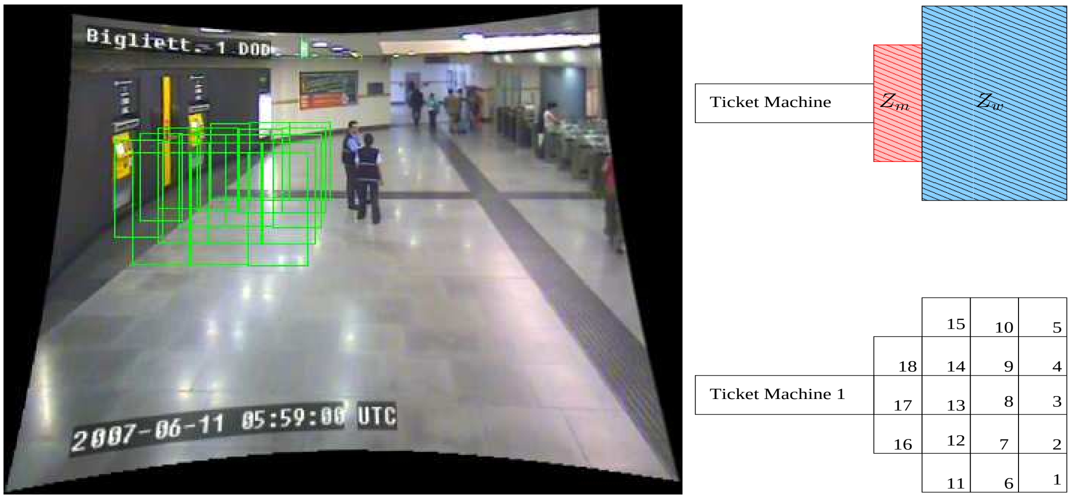

Figure 3: Bounding boxes set-up. On the left, boxes positions for one ticket machine and one camera view; on the right, schematic top-view of the boxes positions and numbering, with the $\mathbf{z}_m$ and $\mathbf{z}_w$ zones definition.

computed. This percentage can be interpreted as the correlation between the foreground image and the binary mask template defined by the box. Integral images [4] are used to speed up computing. These correlations will be refered by $b_i^j(t)$, the correlation in box $1 \leq i \leq 36$, in camera $1 \leq j \leq 2$, at time instant t. Box indices runs from $i = 1 \dots 18$ for machine 1, and from $i = 19 \dots 36$ for machine 2.

This set of features are the core features from which all the features used to recognize the primitive events will be defined. We now define the feature vector for each of the primitive events. This feature vector is computed at each time instant[1].

Machine occupancy features: Only the 3 boxes next to the machine are considered. The feature vector F_{mo} is of dimension 18 and is defined as:

$$F_{mo}(t) = \{b_i^j(t), b_i^j(t+1), b_i^j(t+2)\} \quad 1 \leq i \leq 3 \quad 1 \leq j \leq 2$$

The values are taken at three consecutive time instants, to take into account temporal variations, and increase robustness with respect to ambiguous cases, like someone walking in front of the machine.

Leaving/entering features: All the boxes are used, and the feature vector has 108 dimensions:

$$F_{le}(t) = \{b_i^j(t), bv_i^j(t+1) - bv_i^j(t), bh_i^j(t+1) - bh_i^j(t)\} \quad 1 \leq i \leq 18 \quad 1 \leq j \leq 2$$

where

$$bv_i^j(t) = b_i^j(t) - b_{i+v(i)}^j(t) \quad bh_i^j(t) = b_i^j(t) - b_{i+h(i)}^j(t)$$

with $h(i)$ and $v(i)$ are respectively one of the nearest box in horizontal and vertical directions on the ground plane for box i, for instance, with the numbering of figure 3, for box 1 $v(1) = 2$, $h(1) = 6$, and for box 5 $v(5) = 4$, $h(5) = 10$. Temporal and spatio-temporal derivatives are included in F_{le}, to account for people moving through the zone $\mathbf{z}_m \cup \mathbf{z}_w$, and trying to capture the spatial and temporal characteristics of people leaving or entering this particular zone.

[1]Videos from Turin have a frame rate of 6 fps.

Waiting people features: In that case, we have one feature vector per box. This contrasts with the previous features, which were computed on the whole space, i.e. using all boxes. The vector is of dimension 10, and is defined as:

$$F_w^i(t) = \{m_i(t), m_i(t+3), m_i(t+6), |m_i(t) - m_i(t+3)|, |m_i(t+3) - m_i(t+6)|\}$$
$$with \quad 1 \le j \le 2 \text{ and } m_i(t) = \frac{1}{3}\sum_{k=t}^{t+2} b_i^j(k)$$

This feature vector tries to capture medium-term temporal variations, with mean values at uniformly sampled time instant, and their temporal variations.

3.2 Recognition models

Machine occupancy models: A classification algorithm is applied on the feature vector $F_{mo}(t)$, to classify each frame into machine occupancy or not. Support Vector Machines (SVM) [3] have been chosen because they are known to handle well large dimensional input spaces, and potentially small amount of training data, as it is our case here. SVMs have proven to be successful in solving machine learning problems in computer vision, and other non-linear classification tasks.

In this work, we use a SVM with soft-margin and a Gaussian kernel ($K(\mathbf{b}_1, \mathbf{b}_2) = \exp^{-\frac{|\mathbf{b}_1 - \mathbf{b}_2||^2}{2\sigma^2}}$), whose parameter (the standard deviation σ) was selected on the training data through cross-validation. The cost parameter C, which allows the trade-off between training errors and the soft-margin width, is also set through cross-validation, by an exhaustive search of the parameter space.

An alternative approach has also been investigated using a different model, namely HMM. The model is composed of 2 states, one representing machine occupancy, the other representing all the other events. The probability distribution of the observations' emission is modeled by a Gaussian mixture Model (GMM) with 2 mixtures. In order to impose a minimum duration to each recognized class event (occupied or not-occupied), each state is repeated ten times, which enforces a minimum duration of around 2 seconds (the recording frame rate is 6 frames/s). Note that the repeated states share the same emission probability distribution, i.e. $p(y|q_{i,j})$ are identical for all $j = 1 \dots 10$. The overall state transition model of the HMM is shown on figure 4. The Torch library [1] is used both for HMM and SVM.

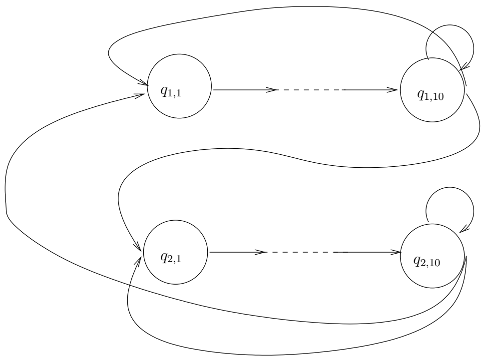

Figure 4: HMM for machine occupancy. States $q_{1,1}$ and $q_{2,1}$ are duplicated to enforce a minimal duration. States $q_{1,j}, j = 1 \dots 10$ share the same emission probability model.

Leaving/Entering model: The same SVM modelling as for machine occupancy is used. Only features and ground truth differ. Parameters have also been set by performing an exhaustive search of the parameter space, and using cross-validation.

Waiting people model: A SVM is also used for detecting this primitive event. However in that case, there is one feature vector per box. The training and testing phase are different from above. In contrast with the previous cases, training is not done on the same set of boxes as for testing. Ground truth is composed of positive examples (people waiting) and negative ones (any other thing) that can be at any location in the scene. One requirement is however that the bounding box used for training is of the same size as the testing box.

Note that since there is one feature vector per box, we will have as many outputs as the number of boxes in the testing phase.

4 Modelling queuing events

4.1 Baseline

We first propose a simple approach to model queuing events, where raw features are directly used. The architecture of the model is shown on figure 5. States 1, 2 and 3 form a left-right model which is the sequence of states we have to go through to recognize a queue event. States 4, 5 and 6 act as a world model, describing data of all other types of events. Observations are continous and their distribution is modeled by a GMM with 4 mixtures.

The features used are the same than those defined for the leaving/entering event, $F = F_{le}$. These features have been chosen because they are defined on the whole space $\mathbf{z}_m \cup \mathbf{z}_w$, and include temporal and spatio-temporal information that are likely to be useful to detect our pattern of interest.

4.2 Layered Model

The layered approach is described on figure 2. The layer approach allows a careful modelling of the event of interest, by defining the primitive events, which can be trained independantly, thus producing a modular architecture. Primitive events can thus be added or removed easily from the model. This approach is also known to require less training data than the standard HMM, with equal performance [6, 2]. The second layer is as well quite robust to changes in the raw features, since it only depends from abstracts events, which are supposedly quite stable. It is thus possible to use the model of the second layer in other situations, where only the first layer has been retrained.

Features: The idea is to take as input feature vector the outputs of the SVM classifiers of the primitive events $y = [F^{MO}_{SVM}, F^{L}_{SVM}, F^{E}_{SVM}, F^{W}_{SVM}]^T$, (where MO=Machine Occupancy, L=leaving, E=Entering, W=Waiting).

In order to have homogeneous values between the SVM scores, a normalization procedure is applied on each output y_i. A median filter of size 3 is first applied to smooth the signal, and the sigmoid function $g(y_i) = \frac{1}{1+e^{-\lambda(y_i - t_i)}}$ is then applied to obtained normalized values, where t_i is the threshold that maximizes the F-measure on the training set of each sub-event i.

A little more work has to be done for the waiting feature. The classifier is applied on each box of the zone $\mathbf{z}_w$, we thus have as many outputs as number of boxes in the zone $\mathbf{z}_w$, and have to combine them in some way. Several ways of combining the outputs of all boxes have been investigated. First, all outputs are normalized using the procedure mentionned above, using $t = 0$, since no ground truth is available for this sub-event. Three sub-zones have been defined in zone $\mathbf{z}_w$ (see figure 6), in which we either take the mean or the max of the normalized SVM score of each box. This thus produces an output vector of 3 dimensions. and $F^{W}_{SVM} = max[F^{W}_{SVM}(space1), F^{W}_{SVM}(space2), F^{W}_{SVM}(space3)]$

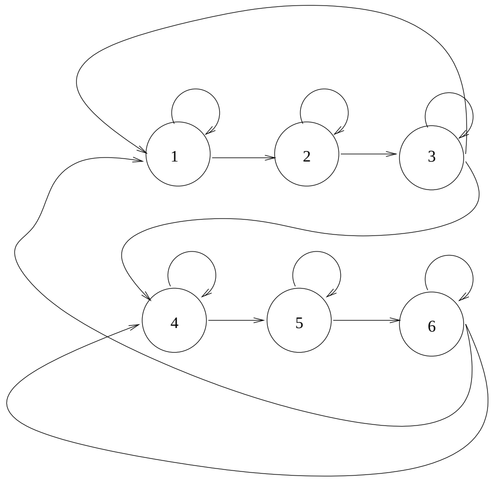

Figure 5: HMM model for queue detection

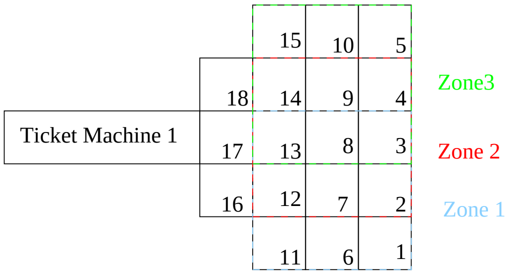

Figure 6: Definition of the 3 static zones in zone $\mathbf{z}_w$ of machine 1. Similar zones are defined for the second machine.

Experimentally, the max operator seems a better choice, since taking the mean value of the SVM outputs of all boxes of the sub-zone seems to produce less detections. The max operator is supposed to be used in the following of this deliverable. Note that in that case F^{W}_{SVM} is also equal to taking the max in zone $\mathbf{z}_w$.

The four normalized features are plotted on figure 7.

Recognition model: The normalized feature vector $g(y)$ is used as input to an HMM, whose architecture is presented on figure 5. States 1, 2 and 3 form a left-right model which is the sequence of states we have to go through to recognize a queue event. States 4, 5 and 6 act as a world model. Observations are continous and the emission probability distribution of the observations is modeled by a GMM with 4 mixtures. Note that this is the same model as the baseline, only the inputs are different.

5 Results

5.1 Datasets and performance measures

The evaluation has been conducted on two videos from Turin, DOD station, in which two ticket machines are visible. The machines are named M1 and M2.

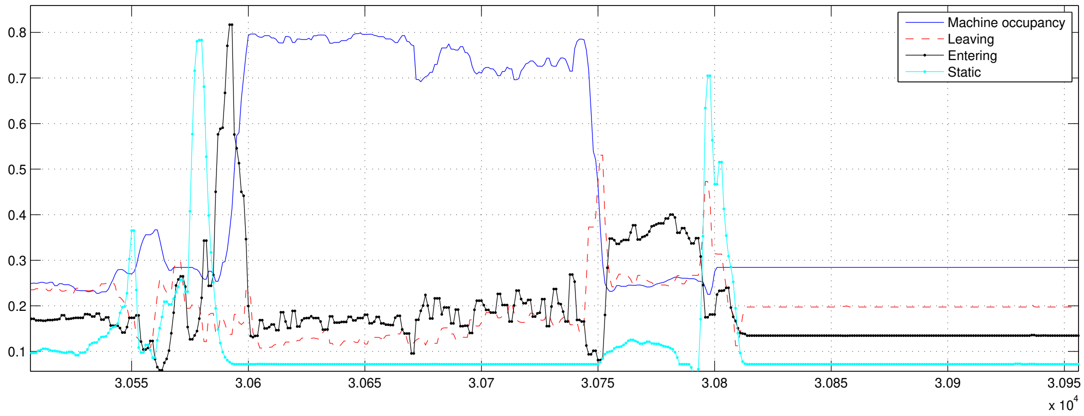

Figure 7: Outputs of the SVM classifers after normalization. The succesion of the events Entering, Occupancy and Leaving can be seen.

- Torino_0 45000 frames (2h05), 2 camera views.
- Torino_1 36000 frames (1h30), 2 camera views.

For all primitive events as well as for queues, the number of frames in which events are present (positive frames) is very small (1 to 5% of the total stream). Consequently, a global frame-based classification rate, which counts if each frame is well classified or not, has no real meaning, since a classifier who would class all frames as negative would get 95 to 99% of good classification. Instead, measures are computed **only on the positive events**, e.g. we count all the frames correctly detected as a positive event N_g (good detection), all the negatives frames detected as positive N_f (false positive), and all the positive frames detected as negative N_m (missed). Using these measures, precision, recall, and F-measure can easily be computed by:

$$Precision = \frac{N_g}{N_g + N_f} \quad Recall = \frac{N_g}{N_g + N_m} \quad F = \frac{2pr}{p+r}$$

We also give an "event" measure, based on the detection of the event itself. An event is a set of contiguous positive detections, which are aligned with the ground truth using dynamic time warping. For an event to be correclty detected, it must share at least one frame with the ground truth. This is complementary to the frame-based measure, and gives a more intuitive results: how well are the events are detected or not. Note however that the matching of events only allows one-to-one matching. For instance, a detected event that spans two ground truth events will produce only a recall of 0.5 even if in frames the recall rate is 1. The frame-based measure then indicates how much of the event is detected. In our view, the event-based F-measure is the most sensible measure to look at.

5.2 Machine Occupancy

Machine occupancy has been evaluated only on the Torino_0 dataset, which has been cut into two sets: a training set of 25000 frames and a test set of 20000 frames. The Torino_0 dataset contains 92 events of machine occupancy, 50 for machine1, and 42 for machine2, approximately uniformely distributed throughout the video. In the SVM case, the training set has also been used (with cross-validation so as to use the whole trainng set) to find the best parameters (σ, C) by an exhaustive search of the parameter space. An example of SVM soft output, and the corresponding hard decision can be seen on figure 8, along with the ground truth. To produce the hard output, the soft SVM output must be

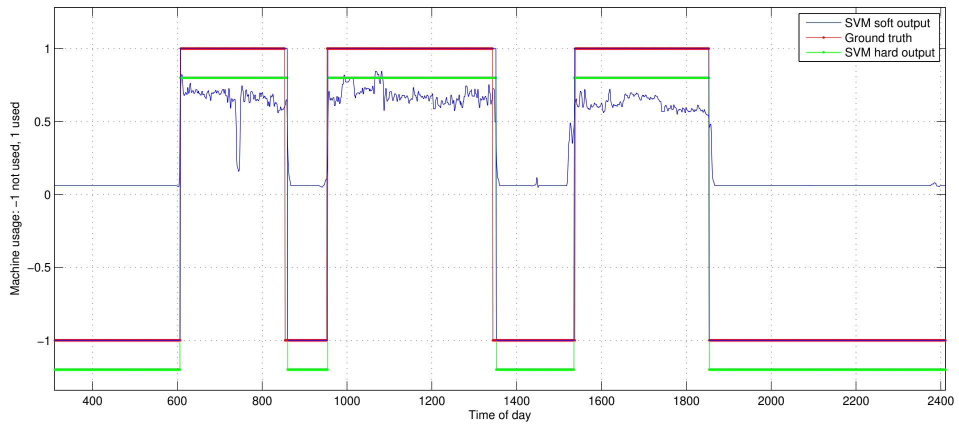

Figure 8: Soft and hard SVM output and ground truth. Zoom on several detections of machine occupancy event.

thresholded. The threshold value chosen is the one that maximizes the event-based F-measure. Recall and precision curves, obtained by varying this threshold are shown on figure 9

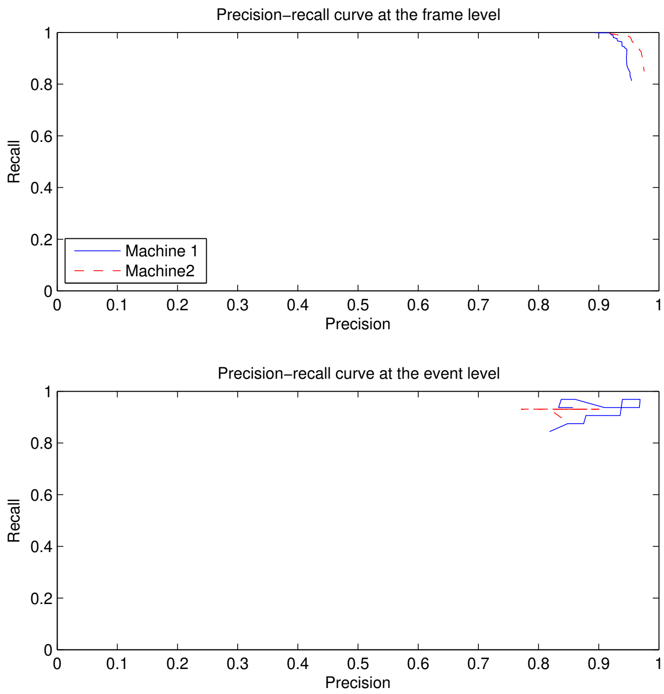

Figure 9: Recall and precision curves for the SVM model for machine occupancy, for both machines.

Both the results of the SVM and HMM approach are presented in table 1, for both ticket machines.

The result given in this table for the SVM is the hard decision, taken as explained above.

	Frame			Event		
	P	R	F	P	R	F
SVM M1	91.8	99.8	95.6	94	96.9	95.4
SVM M2	94.8	98.8	96.7	93.1	93.1	93.1
HMM M1	84.6	99.5	91	90.5	95	92.7
HMM M2	87.5	100	93.3	78.6	91.7	84.6

Table 1: Machine occupancy results for SVM and HMM methods, for 2 different ticket vending machines (M1 and M2).

While detecting machine occupancy is not very difficult, we still have to cope with people walking in front of the machine, and not using it. Results are also dependant on the quality of the background subtraction. Very good results are obtained here thanks to the 2-cameras set-up. Note however that the SVM approach is clearly superior to the HMM-based one, its F-measure being over the HMM one by 2 to 8%.

5.3 Leaving and entering the machine zone

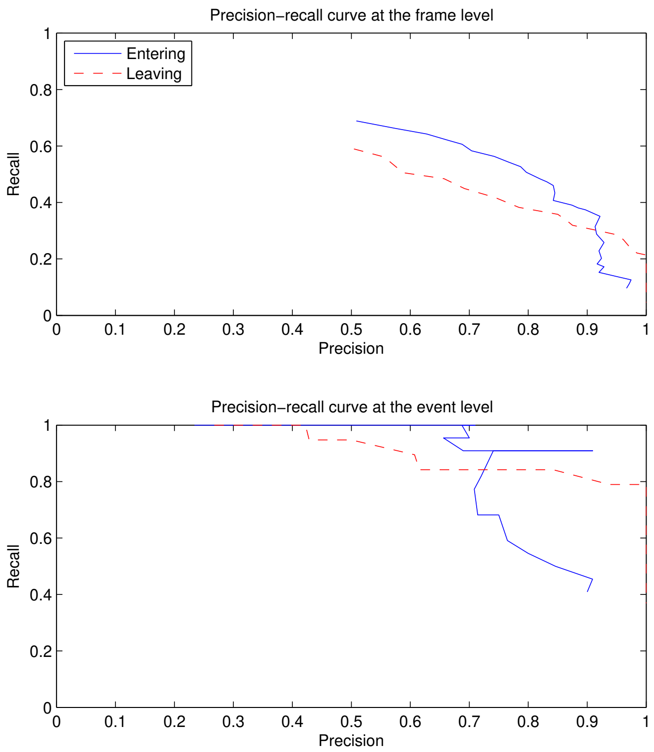

Figure 10: Recall and precision curves for the Leaving and Entering sub-events. Only results from machine 1 are presented here.

The leaving and entering sub-events are evaluated on the same dataset than the machine occupancy, with thus also 25000 frames for the training set, and 20000 frames for the test set. 110 leaving events (52 in Machine 1, 58 in Machine 2) and 110 entering events (57 in Machine 1, 53 in Machine 2) are present in the dataset. The percentage of frames in the stream that contain an event is only 1.5% (both for leaving and entering events), so very few frames are available for learning.

Recall and precision curves for one machine (M1) are given in figure 10. As previously, the soft SVM output is thresholded. The threshold value chosen is the one that maximizes the event-based F-measure. Results for both machines using this threshold are given in table 2. Results in terms of

	Frame			Event		
	P	R	F	P	R	F
Entering M1	74.2	57.3	64.7	87	90.9	88.9
Entering M2	87.1	46.6	60.7	92.3	85.7	88.9
Leaving M1	66	49.1	56.3	93.7	78.9	85.7
Leaving M2	60.9	60.5	60.7	1	70.6	82.7

Table 2: Leaving and entering results for 2 different ticket vending machines.

Figure 11: Output of the SVM classifier for the leaving event, and the associated ground truth.

events are quite good, i.e. most leaving and entering events are detected. False alarms and missed detections are mainly due to the inherent difficulty of defining what a leaving or entering event is, i.e. people wandering around the zone, or entering/leaving it slowly step by step. However, the frame-based measure indicates that frames corresponding to this event are not fully recovered, as can be seen on figure 11. As the detection has a peaked form, most frames are not retrieved.

5.4 Waiting people

Unfortunately no ground truth is available to evaluate this primitive event. It has been evaluated visually, see figure 12. Experimentally, most waiting people seems to be detected. However, currently, the results do not seem to be very satisfying: several false alarms occur, due to people moving slowly, or merely wandering through the waiting zone $\mathbf{z}_w$. Improvements need to be done in the future.

Figure 12: Example of static person detection.

5.5 Queue Detection

For detecting queues, the dataset had to be expanded since very little examples of queues are present in the corpus, see Table 3. The training set has been chosen as Torino_0 (45000 frames), while the test set is Torino_1 (set3, 36000 frames). The training and testing set are in this case from different video files, recorded on different days.

	Machine 1			Machine 2			Total
	#events	mean length	%	#events	mean length	%	
Torino_0	9	25 s	3%	11	40 s	5.7%	20
Torino_1 set 1	2	15 s	0.5	2	30 s	1%	4
Torino_1 set 2	0	-	0	0	-	0	0
Torino_1 set 3	8	42s	5.6%	5	8s	0.7%	13

Table 3: Ground truth for queues on the Torino_0 and Torino_1 datasets.

	Frame			Event		
	P	R	F	P	R	F
M1	11.5	100	20.6	11.7	87.5	20.6
M2	2	100	3.8	1.3	40	2.5

Table 4: Results of the baseline method for detecting queues.

Results of the baseline approach are shown in table 4. These results are very low, with a very high number of false alarms. The model is not able to distinguish between a queue event happening, and the mere presence of people in the zone. Note that the frame-based recall is 100% but the event-based recall is lower. This can be explained by the fact that several queueing events are detected as a single event only, thus only one event is counted as correctly retrieved. This happens especially on machine 2. Results are especially low on this machine because very few events are present in the test set, and the model might be disturbed by a lot of people moving around or being about the zone, while not actually queuing. For both machines, false alarms are also due to a technician repairing the machine, with a huge toolbox behind him, this can be seen on figure 14.

	Frame			Event		
	P	R	F	P	R	F
M1	54.3	99.9	70.4	33.3	100	50
M2	6	100	10	10	80	18

Table 5: Results from the layered HMM approach to detect queues.

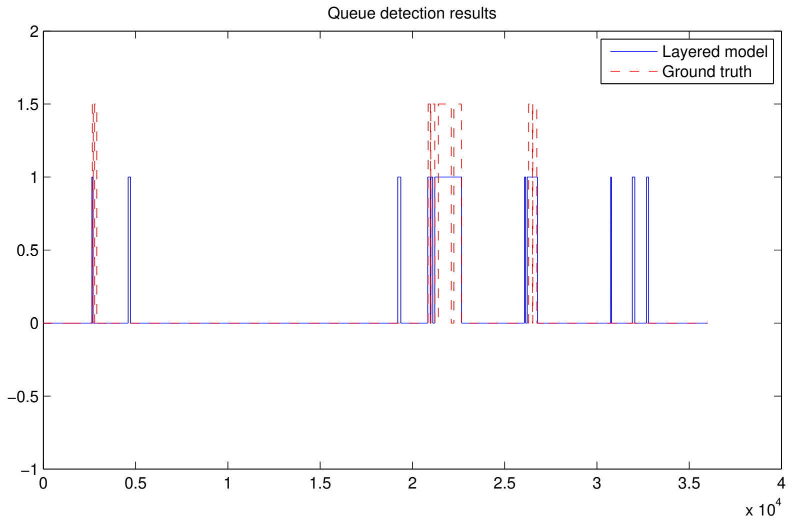

Figure 13: Classification results of the layered model for queue detection on the whole test set (2h).

In the layered approach, results are better than in the baseline, although quite a high number of false alarms are still present. Most false alarms are occuring when several people are present in the waiting zone $\mathbf{z}_w$. Classification results for machine 1 are shown plotted on a timeline in figure 13, where several false alarms are visible. One of these (the one occuring around frames 19200/19250) is illustrated on figure 14, where 2 people are indeed behind one another, but no queue is happening. The man behind is actually enquiring about something. Similar to the baseline approach, results of machine 2 are very low, with a very high false alarm rate.

6 Conclusion and perspectives

We have presented an event recognition based approach for monitoring ticket vending machines. A model for detecting high level events with a layered architecture has been presented. The high level event is composed of several sub-events, which are detected by discriminative models (SVMs). These sub-events are also of interest and can provide useful information to monitor the station's activity. Classification results for the queuing event show that the layered approach outperforms a baseline one with raw features, but still need improvement. However, the approach does not require a very large training set and has a low computationnal load.

Improvements are still necessary for the waiting people sub-event, as well as for the layered model. A more constrained architecture of the HMM could help to actually discriminate the queue events.

Figure 14: Example of a false alarm: technician repairing the machine, plus someone watching/enquiring.

Another possibility is to use also a discriminative model for the second layer of the layered model.

References

[1] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. Technical report, 2002.

[2] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.*, 96(2):163–180, 2004.

[3] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[4] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[5] Jian Yao and Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *Proc. CVPR Visual Surveillance workshop (CVPR-VS)*, Minneapolis, June 2007.

[6] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, Iain McCowan, and Guillaume Lathoud. Modeling Individual and Group Actions in Meetings: a Two-Layer HMM Framework. *IEEE Transactions on Multimedia*, 8(3):509–520, 2006.