Shopping scenarios semantic analysis in videos

R. Sicre & H. Nicolas
LaBRI, University of Bordeaux, 351 Cours de la libération, 33405 Talence Cedex, France
MIRANE SAS, 16 rue du 8 mai 1945, 33150 Cenon, France
sicre@labri.fr - nicolas@labri.fr

Abstract

This paper presents a computer vision system that analyzes a scene, recognizes human behavior, and produces a semantic interpretation of the recognized behavior. Scene analysis consists of detecting motion and tracking detected objects. Then, the system detects events that define the current behavior of a person. Once behaviors are classified the system recognizes various predefined scenarios. Finally sentences are generated to describe each person's action in the scene.

Our research is developed in order to build a real-time application that recognizes human behaviors while shopping. In particular, the system detects customer interests and interactions with various products of a point of purchase.

1. Introduction

Computer vision applications are developed in various fields of applications: surveillance, traffic monitoring, video games, marketing, etc.

In the field of marketing, digital media is used extensively on point of purchase. Originally playing advertising clips one after another, media impact was lower than expected. It is then of primarily concern to identify ideal content and location for this media to maximize its impact on customers. Nowadays a few software systems exist. Some of them track customers to obtain statistical information regarding their habits, where others calculate directly the media audience, using face detection.

The study introduced in this paper is along the same lines and aims at adapting the clips to the behavior of the customers in order to improve the digital media impact. Specifically, we detect customers picking up products from known areas in real-time using a fixed camera. The detection of such an event results, for example, in playing a clip related to the product.

After a short review, we present the system: first the scene analysis, then the behavior analysis, scenario recognition, and semantic interpretation. We show some results and conclude with future work.

2. Previous work

This section presents different techniques for behavior analysis and semantic description of behavior. For an overview of the field, the reader can refer to various surveys [10] [3].

2.1. Behavior analysis

We are interested in observing humans as deformable objects. The goal is to analyze and recognize motion samples in order to draw high level conclusions. This analysis is usually made in two steps, description and recognition of specific actions. The first step is to define a model that describes each possible action. Once the training is computed, there are several methods that can classify the data like Dynamic Time Warping, HMM and Neural networks [8]. However, we use Finite State Machines (FSM) [2]. This logical method primarily advantage is to avoid the training phase.

2.2. Semantic description of behavior

Many applications require a description of object behavior in natural language, suitable for non-specialist operator. There are two main categories of behavior description methods. Statistical models, like Bayesian network models, interpret events and behavior as interactions between objects by the analysis of time sequences and statistical modeling [6]. Formalized reasoning represents behavior patterns using symbol systems then recognizes and classifies events with reasoning methods [4]. We choose formalized reasoning for its simplicity.

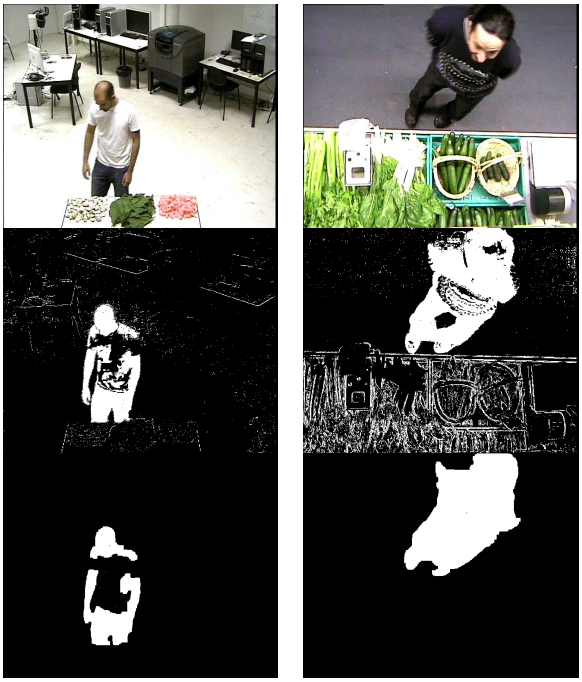


Figure 1. Motion detection on videos from the datasets LAB1 (left) and MALL1 (right). The first row corresponds to the video frame, the second row is the rough motion detection, and the last row is the result

3. Scene Analysis

In order to detect and analyze behaviors, the system needs a scene analysis. We use existing methods that can be divided in two phases: first, motion detection finds moving regions that do not belong to the background. Then, these regions are tracked over the frame sequence. Fast methods were selected in order to cope with the real-time constraints of our application.

Motion detection uses a pixel based model of the background [7]. A mixture of Gaussians is associated with each pixel, in order to characterize the background. This model is updated on line. A Gaussian distribution is matched to the current value of each pixel. If this Gaussian belongs to the background, the pixel is classified as such. Otherwise, it is considered as foreground, see figure 1.

Morphological filters are applied on this result. However, a detected person can be covered by several disconnected regions, because the algorithm misses part of the person, see figure 1 first column. Thus, we need to merge regions. This merging process is based on the distance and relative position of the regions bounding boxes. If the algorithm can not clearly decide to merge the regions, the tracking module makes the final decision using temporal information.

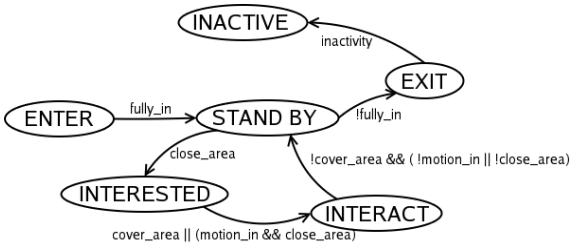


Figure 2. Part of the Finite State Machine, all transitions are not written to make it clear.

To track regions over the frame sequence, the system calculates for each of them a descriptor based on its position, size, surface, first and second order color moments. These descriptors are first matched from one frame to the next, using a voting process, then matches are verified using previously tracked objects.

4. Behavior Analysis

In this section, we build a model that represents people's behavior. We create various states that correspond to the current behavior of a tracked person. The chain of state describes a scenario that the person plays. Each state is detected based on measurement in the video. We know interest areas where products heap are located. Six states are used in a finite state machine (FSM) [9] and are defined as follow.

- **Entering**: A new person appears in the scene. It occurs when a newly tracked person is connected to an edge of the image.
- **Exiting**: A person leaves the scene. It is calculated the same way as the entering state, for a person already tracked.
- **Interested**: A person is close to a product area. It is detected when a person connects a product area.
- **Interacting**: A person picks up products or is about to do so. This state occurs in one of two ways. Firstly, a tracked person covers a product area with a surface that is larger than a theoretical hand size. Secondly, motion is detected in a product area and a person is nearby. This event corresponds to an object that is moved in the interest area and is probably picked up. However, this event is hard to detect since objects are all the same in the heap.
- **Standing by**: A person is in the scene but not close to any product area or image boundary. We also calculate if the person is moving or stopped
- **Inactive**: A person is not tracked anymore. The system loses track of a person. This event mostly happens when a person exits the scene, or is occluded by something in the scene or another

person. We do not manage multi person occlusions.

The state machine is synchronous and deterministic. Synchronous means that the machine iterates over each new frame. Based on the previous state, the system calculates a new one by testing each transition condition. If a condition is satisfied, the system moves to the new state. Otherwise, the system stays in the same state. The machine is deterministic because for each state, there can not be more than one transition for each possible input.

We save the path of each tracked person in the FSM in order to interpret a high-level scenario. Figure 2 shows a possible path through the FSM.

5. Scenario recognition

This section aims at defining and recognizing specific high-level scenarios. For our application, it is interesting to detect various scenarios that describe different customer behavior. Depending on the scenario occurring, the media communicates in a different way. In particular, the system differentiates people passing by from people interested or interacting with products.

The FSM is a method that allows determining a state based on the previous state and the sensors input. This is the reason why, we use Allen's theory that allows us to define more complex scenarios with extra temporal relations between states.

5.1. Scenario definition

Allen's interval algebra [1] is a calculus for temporal reasoning. It defines possible relations between time intervals and can be used for reasoning about temporal descriptions of events. Specifically, if we define two events, *A* and *B*, that last for a period of time, Allen's theory determines if *A* happens *before B*, *during B*, *meets B*, *overlaps with B*, *starts B*, *finishes B*, is *equal* to *B*, and all the inverses of theses relations.

In our approach, we use these conditions on the states of the FSM. Each state is an event that last for a certain interval of time. However, we can not use all the possibilities of Allen's theory, because all the states are mutually independent in the deterministic FSM. Therefore, we use only relations like *before*, *meet*, and their inverse *ibefore* and *imeet*. Here are three different scenarios that are recognized:

1. Person walking by
2. Person walking by and being close to a products
3. Person walking by and interacting with products

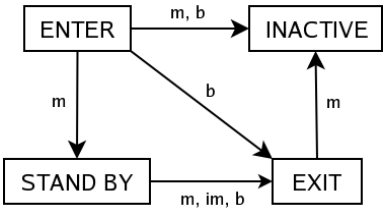


Figure 3. First scenario represented as a network using Allen's theory.

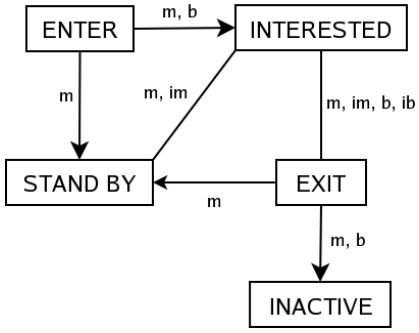


Figure 4. Scenario 2.

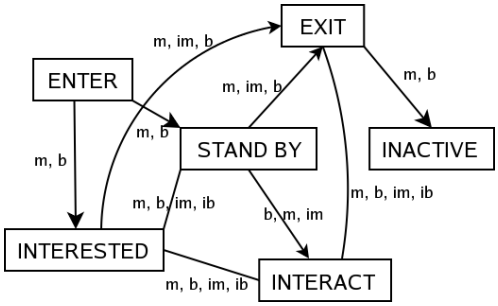


Figure 5. Scenario 3.

The first scenario is simple and can be thought of as a sequence of states: Enter – Stand by – Exit – Inactive or Enter – Exit – Inactive. Using Allen's theory, we represent these scenarios as networks [5] shown in figures 3, 4, and 5, respectively for the three predefined scenarios. *m* and *b* on the transitions represent the relations *meet* and *before*. *im* and *ib* are the inverse relations. Note that *A meet B* means that the event *B* starts as soon as event *A* finishes.

5.2. Scenario recognition

The next step is to recognize the three scenarios. For each tracked person, we build the three networks. As a person moves from one state to another in the FSM, we check the validity of each transition in the networks. As soon as two scenarios are no longer valid, the system prints a sentence suggesting that the person plays the only valid scenario. Once the person leaves the scene,

or is “Inactive”, we check the validity of each scenario. Then the system prints the more likely scenario that was followed by the tracked person, as well as a validity score. It is interesting to note that small detected object or object that were detected on a few frames are likely to correspond to noise. Therefore the system does not print information regarding these objects.

6. Semantic interpretation

This section aims at describing person’s behavior in natural language and is based on the previous analysis. The generated sentences summarize the current state of action of a person. Expressing human activities is accomplished using case frames [4]. This tool links cases to sentences in natural language. We use simple case frames composed of three categories as illustrated in the following example:
AG: “Person 1”, PRED: “interacts with”, LOC: “area1”

AG, PRED and LOC are the agent, the predicate, and the locus of the described action respectively. These three cases allow us to describe all the scenarios we look for.

Sentences are built using a hierarchy of actions that is composed of case frames to represent each node. A parent node derives its children by redefining of the verb (predicate is modified) or the locus (locus is modified) as shown in figure 6. In particular, by going through the hierarchy of action, the case frame is refined until a final node is reached. Final nodes do not have children and contains all the components of the sentence we want. Here are the final node predicates:

is walking around - is stopped - enters the scene - exits the scene - is interested in - interacts with - is gone

However, in practice, to follow rapidly changing states, we need a dynamic model. We use each type of verb to build different states as well as each locus and construct a State Transition Diagram (STD), see figure 7. As most of the analysis is completed in the previous phase, the hierarchy of action is simplified.

We explain in the algorithm that follows how case frames are generated from the STD.

1. Let *s* be a current state. For a newly estimated position of the tracked person, semantic primitives are evaluated downward from the top of the STD to determine the new state *s’*.
2. If state *s* and state *s’* are identical, no case frame is generated, and we go back to step 1.
3. Otherwise, a case frame associated to the new state *s’* is generated, and we go back to step 1.

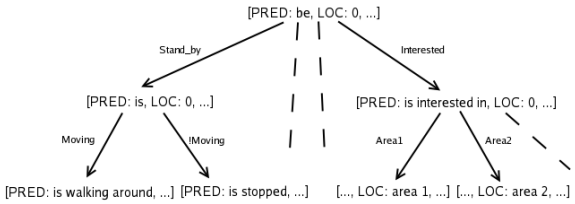


Figure 6. Sample of the hierarchy of actions.

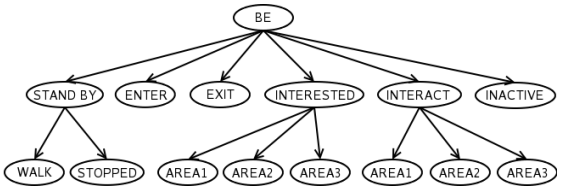


Figure 7. State transition diagram.

More specifically, when a state change occurs, a case frame is generated, and a sentence is printed.

7. Results

7.1. Data description

We use different datasets taken with the same camera, see table 1. A part of the datasets was taken in our laboratory (LAB1, 2, and 3). The others were taken in a real shopping mall (MALL1 and MALL2). The two first datasets (LAB1 and MALL1) own five and six sequences respectively with two and four actors and contains a lot of interactions with products, see figure 1 and figure 10. LAB2 is a dataset where there is no direct interactions with products and has three different actors, see figure 8. Objects or products taken by people have different shapes, colors, and sizes in the sequences of video. Also, all products are identical in the heaps. LAB3 and MALL2 are two datasets where multiple people interact together. Two to four people interact simultaneously.

7.2. Semantic analysis

The behavior analysis and semantic interpretation give good results for various situations, see figure 8, 9, and 10. However there are a couple of limitations. The main limitation is due to the scene analysis phase. Although the system’s motion detection and tracking give good results, the merging process has limitations, when many people are close or occluding each other. They can be merged as one single object, especially if they are static. Also, the “Entering” and “Exiting” states are not always detected because the person is not entirely detected, see figure 9 frame 12.

7.3. Scenario recognition

Scenario recognition offers interesting results. For each video of the datasets MALL1, LAB1, and LAB2, the scenarios played by the tracked persons are correctly identified.

We detect the scenarios in two different ways. First we detect scenarios on-line, and suggest a scenario as soon as the two others are invalid. It is interesting to note that this phase does not always offer perfect results, see figure 9 and 10. The second method checks the entire path once the person leaves the scene and look for the closest scenario. This second process offer correct results for the videos tested. Table 1 shows the percentage of correct state transitions during each video.

7.4. Test on datasets

We run tests on various sequences of video in order to understand the system's behavior in various cases, see table 1. As the system generates a state for each detected object, for each frame, we compare these results to the ground truth, which was generated by hands. Then, we calculate the percentage of correct states, as well as the precision and recall for the "Interact" state.

The dataset LAB2 has no interaction with products in the sequences. This dataset has better results than the LAB1, and we can conclude that "Interact" is less precisely detected than the other states, for the videos taken in the LAB.

The dataset LAB1 give better overall results than MALL1. This difference is mainly due to noises that are more significant in MALL1 as well as slight camera motion during the capture. However, MALL1 performs very well on recall and precision due to the position of the camera, located directly above the products and closer to them than on LAB1. We understand that the camera location is really important to maximize the precision of the system. A position close to the products performs better on "Interact" detection. However, having the camera too close to the products make us lose information about the customers, since they are only detected when they are near the products.

Even if the system does not manage occlusions of people, the detection of interaction in multi-person datasets works as well as in the first datasets.

Finally, we note general recommendation regarding products. Small products are harder to detect, when they are picked up. Also, bright objects tend to generate false detection as a person is around, due to its shadow.

7.5. Computation time

Finally, the application has to generate responses quickly as soon as a specific event is detected. The program is tested on a computer with a Pentium M, 1.73 Ghz and 500 Mb of RAM. The application can analyze 6 to 10 frames per second for an image resolution of 704x576 or 640x480 respectively.

8. Conclusion

This paper presents a novel type of application, using computer vision in the field of marketing. The system tracks and analyzes behaviors of shoppers, then recognizes predefined scenarios regarding their actions, interests, and interactions with products.

The system is tested and offers interesting results, 73% of the frames are correctly labeled, for sequences taken in a real environment. Interactions with products are well detected with a precision of 0.81 and a recall of 0.9. Scenarios are precisely detected with 98.6% of valid transitions. This evaluation helps us understand how the system behaves, as a prototype will soon be tested for a long period of time. As future work, the method can be improved to cope with occlusions.

9. References

[1] J. F. Allen, *Maintaining knowledge about temporal intervals*. Communications of the ACM, 1983.

[2] F. Bremond, G. Medioni, "Scenario recognition in airborne video imagery", Proc. Int. Workshop Interpretation of Visual Motion, pp 57-64. 1998.

[3] W. Hu, T. Tan, L. Wang, S. Maybank "A survey on visual surveillance of object motion and behaviours," IEEE Transaction on systems, man, and Cybernetics, pp 334 – 352, 2004.

[4] A. Kojima, T. Tamura, K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions", Int. J. Comput. Vis, vol. 50, no. 2, pp 171 – 184, 2002.

[5] C.S. Pinhanez, A.F. Bobick "Human action detection using PNF propagation of temporal constraints", *Proc. Conf. on Computer Vision and Pattern Recognition*, 1998

[6] P. Remagnino, T. N. Tan, A. D. Worrall, and K. D. Baker, "Multi-agentvisual surveillance of dynamic scenes," Image Vis. Comput., vol. 16, no.8, pp. 529–532, 1998.

[7] C. Stauffer, W. Grimson, "Adaptive background mixture models for real-time tracking" Proc. Computer Vision and Pattern Recognition, pp. 246–252, 1999.

Table 1. This table represents the percentage of correctly detected states, the precision and recall for the “Interact” state, and the percentage of correct transition for the scenario recognition. Measurements are calculated for each frame on videos in various datasets. Videos have 15 frames per second.

Dataset	Video	Frames	Correctness	Precision	Recall	Scenario
MALL1	1	327	70,03%	0,9005	0,795	98,65%
	2	444	74,77%	0,7692	0,9184	98,68%
	3	434	66,13%	0,5778	0,9512	100%
	4	336	70,83%	0,9326	0,9326	99,65%
	5	164	76,22%	0,8571	0,9231	96,46%
	6	232	79,74%	0,8444	1	98,34%
	mean		**72,95%**	**0,8136**	**0,9021**	**98,63%**
LAB1	1	545	85,87%	0,8558	0,6742	100%
	2	672	74,40%	0,7172	0,6698	98,96%
	3	704	76,28%	0,812	0,662	100%
	4	771	60,57%	0,5349	0,4978	100%
	5	518	92,66%	0,8174	1	100%
	mean		**76,13%**	**0,7475**	**0,7008**	**99,79%**
LAB2	1	476	87,61%	N/A	N/A	95,82%
	2	342	82,46%	N/A	N/A	98,60%
	3	143	91,61%	N/A	N/A	100%
	4	303	95,71%	N/A	N/A	100%
	mean		**89,35%**			**98,61%**
LAB3 MP	1	212	N/A	0,9815	0,9217	N/A
	2	211	N/A	0,9789	0,7209	N/A
	3	300	N/A	0,7643	0,673	N/A
	4	303	N/A	0,7733	0,6591	N/A
	5	259	N/A	0,7311	0,58	N/A
	mean			**0,8458**	**0,7109**	
MALL2 MP	1	215	N/A	0,9595	0,9467	N/A
	2	208	N/A	1	1	N/A
	3	735	N/A	0,7611	0,8566	N/A
	4	153	N/A	0,7429	0,8966	N/A
	5	382	N/A	0,7203	0,7103	N/A
	6	504	N/A	0,9136	0,8862	N/A
	mean			**0,8496**	**0,8827**	


51


Frame 1 : No one is in the scene
Frame 54 : The person 1 is walking around
Frame 101 : The person 1 exits the scene
Frame 115 : No one is in the scene
65


Scenarios :
Frame 101 : Person 1 plays scenario 1
Frame 115 : Person 1 had scenario 1 valid all along
101

Figure 8. Results of the video 3 in LAB2.


14


19

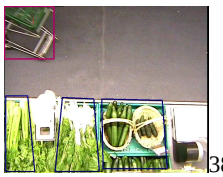

22

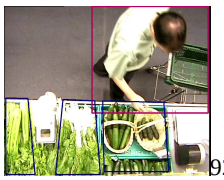

37


41

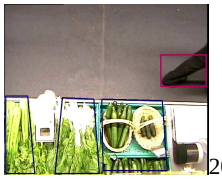
Frame 1 : No one is in the scene
Frame 12 : The person 1 is stopped
Frame 13 : The person 1 enters the scene
Frame 15 : The person 1 is interested in area 3
Frame 16 : The person 3 enters the scene
Frame 19 : The person 3 is interested in area 1
Frame 20 : The person 1 interacts with area 3
Frame 21 : The person 3 interacts with area 1
Frame 37 : The person 3 interacts with area 2
Frame 38 : The person 3 interacts with area 1
Frame 41 : The person 3 exits the scene
Frame 43 : The person 3 is gone
Frame 44 : The person 1 exits the scene
Frame 48 : No one is in the scene
Scenarios :
Frame 12 : Person 1 has no perfectly valid scenario
Frame 21 : Person 3 plays scenario 3
Frame 43 : Person 3 had scenario 3 valid all along
Frame 48 : Scenario 3 was valid for the person 1
Correctness : 33 / 35

Figure 9. Results of a sequence with two persons successfully analyzed.


38


92


149


204

Frame 2 : No one is in the scene
Frame 32 : The person 1 is walking around
Frame 34 : The person 1 exits the scene
Frame 82 : The person 1 is interested in area 3
Frame 93 : The person 1 interacts with area 3
Frame 188 : The person 1 is interested in area 3
Frame 200 : The person 1 exits the scene
Frame 208 : No one is in the scene
Scenarios :
Frame 82 : Person 1 has no perfectly valid scenario
Frame 208 : Scenario 3 was valid for the person 1
Correctness : 178 / 181

Figure 10. Results of the video 6 in MALL1.

[8] D. Tran, A. Sorokin, “Human activity recognition with metric learning”, Euro. Conf. on Computer Vision, 2008.

[9] F. Wagner, Modeling Software with Finite State Machines: A Practical Approach, Auerbach Pbl. ch.4, 2006.

[10] A. Yilmaz, O. Javed, M. Shah, “Object tracking: a survey”, ACM Computing Surveys, 2006.