Human Behavior Analysis at a Point of Sale

R. Sicre and H. Nicolas

LaBRI, University of bordeaux - 351 Cours de la libération, 33405 Talence Cedex, France
Mirane S.A.S - 16 rue du 18 mai 1945 33150 Cenon, France

Abstract. This paper presents a method that analyzes human behavior in a shopping setting. Several actions are detected and we are especially interested in detecting interactions between customers and products. This paper first presents our application context, the advantages and constraint of a shopping setting. Then we present and evaluate several methods for human behavior understanding. Human actions are represented with Motion History Image (MHI), Accumulated Motion Image (AMI), Local Motion Context (LMC), and Interaction Context (IC). Then we use Support Vector Machines (SVM) to classify actions. Finally, we combine LMC and IC descriptors in a real-time system that recognizes human behaviors while shopping to enhance digital media impact at the point of sale.

1 Introduction

Behavior understanding is a growing field of computer vision. Several applications are developed in order to detect human behaviors in various contexts, such as content-based video analysis and indexing, video-surveillance, interactive applications, etc.

Marketing is a new field of applications that uses computer vision systems to measure media, and display, efficiency. The marketing field has evolved lately. The use of digital media, or digital signage, at point of sale becomes more and more popular. This media offers new forms of communication with customers that bring along new issues. For example, media playing advertising clips one after another does not have significant impact on customers. It is then of primarily concern to identify ideal content and location for these media, in order to maximize its impact on customers. Nowadays several software systems help solving these problems. A few systems track customers, in a video-surveillance context, to obtain statistical information regarding customers' habits and displacement inside shopping malls. Various systems calculate directly the media audience and opportunity to see the media, using face detection.

The study introduced in this paper is along the same lines and aims at improving the impact of digital media by maximizing interaction between media and customers. Furthermore, we want to produce statistical data on customers' interaction with products. More 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 video analysis part, then the behavior description and recognition. Finally we show some results and conclude.

G. Bebis et al. (Eds.): ISVC 2010, Part III, LNCS 6455, pp. 635–644, 2010.

2 Previous Work

Human behavior can be identified in many different contexts [5] [18] [12]. The goal of behavior analysis is to recognize motion samples in order to draw high-level conclusion. There are several issues due to the fact that we match real-world activities to outputs perceived by a video processing module. We have to select relevant properties computed with video processing tasks and handle the incompleteness and uncertainty of these properties.

Behavior analysis is generally composed of two steps: description and recognition of actions. Action description selects relevant measurements that characterize various actions in a specific context. Recognition is usually composed of two processes. First labelled data is generated and used as training. Then these training samples are used to recognize actions using learning methods, such as Hidden Markov Model, Neural Networks [17], Support Vector Machine (SVM) [6] [14], etc. However, recognition can be accomplished using a logical model based on the description measurements, such as Finite State Machine (FSM) [7]. Although this method is not very flexible, it can be applied without the training phase.

In our study, we combine FSM and SVM. FSM are used to detect the simple actions and SVM classify interactions between customers and products.

3 Shopping Setting

This section presents the shopping setting with more details. As we see in the previous work, behavior analysis is used in various contexts. Several datasets were used as a baseline for many researchers. We categorize four sorts of datasets used for different applications.

First datasets, like [14] [1] [19], aim at detecting specific motion behavior like people waving, jumping, walking, running, boxing, etc. Videos are mainly taken without camera motion and focus essentially on the actor.

The second type of datasets [9] [16] are directly extracted from movies. These datasets are used to detect people shaking hands, hugging, answering the phone, etc.

Different kind of behavior can be detected in a video-surveillance setting [11] [16], like meetings, language drop, crowd analysis, etc.

The last type of datasets concern sports videos [13]. The configuration varies and can be focused on the actors, extracted from a TV-show, or surveillance like.

Nowadays, only a few papers used dataset coming from point of sale [15] [6]. The shopping setting is between behavior analysis like [14] that observes a person to detect its moving behavior and video-surveillance that detect interactions between people, luggage, specific areas of the scene, etc. Thus we want to detect customer moving behavior as well as interaction with specific areas of the scene, i.e. products areas.

4 Behavior Model

This section presents the model used to define human behavior while shopping. At a point of sale, customers walk around products, look at prices, pick up products, etc.

We create six states that correspond to the current behavior of a person. The chain of states describes the scenario played by the person. The same model is used for higher-level scenario detection and semantic interpretation [15].

- **Enter**: A new person appears in the scene.
- **Exit**: The person leaves the scene.
- **Interested**: The person is close to products, i.e. possibly interested.
- **Interacting**: The person interacts with products, is grabbing products.
- **Stand by**: The person is in the scene but not close to any product area or image boundary. The person can be walking or stopped.
- **Inactive**: The person has left the scene.

5 Video Analysis

In order to detect behaviors, we require information concerning every person in the scene. For every frame, we need people's location, contours, etc. Therefore we use a motion detection and object tracking process. 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 final application. [20] presents a clear overview of the object tracking methods.

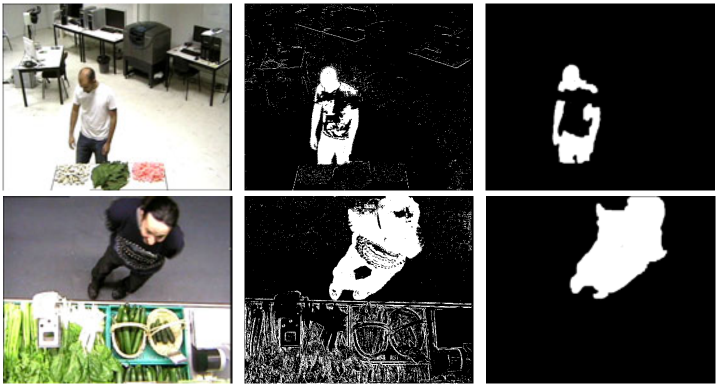


Fig. 1. Motion detection on videos from LAB1 (top) and MALL1 (bottom) datasets. Frames are on the first columns, rough detection on the second and filtered results on the third.

Motion detection uses a pixel based model of the background to generate precise contour of the detected regions. A mixture of Gaussians is associated with each pixel, in order to characterize the background [21]. 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 the pixel is considered as foreground. Morphological filters are finally applied on this result, see figure 1.

Object Tracking is composed of two main processes. First, we calculate, for each region, a descriptor based on its position, size, surface area, first and second order color moments. Then, these descriptors are matched from one frame to the next, using a voting process. Secondly, we use matched regions to build and update the object list. An object is a region that was tracked for several frames. Matched regions are used to update information about objects: location, size, etc. Then unmatched regions are compared to inactive unmatched object to solve miss-detections. We also detect regions split and merge to detect occlusions.

6 Behavior Description

We focus on tracked person's interactions with product areas, i.e. people grabbing products. While grabbing a product, a person first reaches out with its arm, then grasps a product, and finally take the product. These different phases in the "product grabbing" event correspond to observable local motion of the person. Following the idea that similarity between various motions can be identified through spatio-temporal motion description, a corresponding descriptor has to be composed of sets of features sampled in space and time [17] [4]. This section presents various description methods.

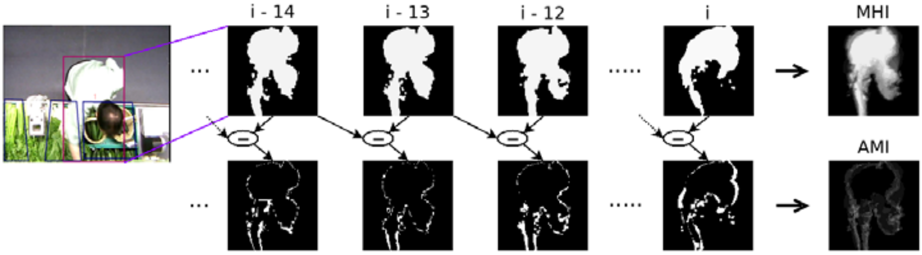


Fig. 2. Diagram showing how MHI and AMI are generate from the frame sequence

6.1 Motion History Image

MHI is a temporal template used as model for actions [2]. MHI offers information concerning a person shape and the way it varies along a local period of time. We aggregate a sequence of foreground object masks, scaled to a standard size of 120x120 pixels, see figure 2. MHI is computed as follows:

$$MHI(x,y) = \frac{1}{T}\sum_{t=1}^{T} I(x,y,t) \qquad (1)$$

Where $I(x,y,t)$ is the pixel value of the Image I at position (x,y) at time t. T is the time interval used to calculate the MHI, we choose $T = 15$.

We define two energy histograms by projecting MHI values along horizontal and vertical axis [8]. These energy histograms are calculated as follows:

$$EH_h(i) = \sum_{j=0}^{W-1} MHI(i, j),\ i = 0,...,H-1$$
$$EH_v(j) = \sum_{i=0}^{H-1} MHI(i, j),\ j = 0,...,W-1 \tag{2}$$

H and W are relatively the height and width of our scaled image. We have $H = W = 120$. These two energy histograms are used as a 240 (120x2) dimensional descriptor to recognize Interactions.

6.2 Accumulated Motion Image

AMI [8] was inspired from MHI and Motion Energy Image (MEI) [2]. As we see in the previous section, MHI and MEI use the entire silhouette. However, only areas including changes are used to generate the AMI that is defined as follows:

$$AMI(x, y) = \frac{1}{T}\sum_{t=1}^{T}\left|D(x, y, t)\right| \tag{3}$$

Where $D(x, y, t) = I(x, y, t) - I(x, y, t-1)$. We note that the image difference is calculated between two scaled masks and we keep $T = 15$, see figure 2.

We calculate the same energy histograms presented in the previous section that are used as descriptor (240 dimensions) to recognize interactions.

6.3 Local Motion Context

We then choose to describe motion using pixel-wise optical flow [4]. Since optical flow is not very accurate, we use histograms of features over image regions. Such a representation is tolerant to some level of noise, according to [17].

Local motion: First, each person's mask is scaled to a standard size of 120x120 pixels, while keeping aspect ratio. Then, the optical flow is computed using Lucas Kanade algorithm [10]. The result of this process is two matrixes with values of motion vectors along x and y axis. We separate negative from positive values in the two matrixes, and end out with 4 matrixes before applying a Gaussian blur to reduce the effects of noises.

Silhouette: A fifth matrix, representing the person's silhouette, is computed from the scaled mask.

Data quantization: We reduce the dimensionality of these matrixes to filter noises and save computation time. Each matrix is divided into a 2x2 grid. Each grid cell gets its values integrated over an 18-bin radial histogram (20 degrees per bin). Matrixes are now represented by a 72 (2x2x18) dimensional vector.

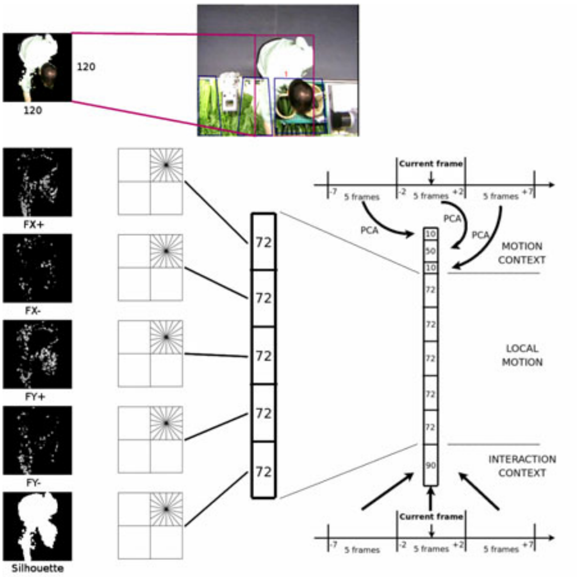


Fig. 3. Diagram representing the local motion and behavior context descriptor same summed

Temporal context: to take into account temporal information, we use 15 frames around the current frame and split them in three sets of 5 frames: past, current, and future. After applying Principal Component Analysis (PCA) on each set's descriptors, we keep the first 50 components for the current set, while we only keep the first 10 components for the past and future sets. The temporal context descriptor possesses then 70 (10+50+10) dimensions.

The final descriptor is composed of 430 (72x5+70) dimensions, see figure 3.

6.4 Interaction Context

This last descriptor is based on interaction with product areas. These areas are assumed to be known. We use six measurements calculated as follow:

- The person's surface covering a product area.
- A Boolean that is true when this covering surface is bigger than a theoretical hand size or when a person is connected to a product area and there is motion detected on this area.
- The surface of the person.
- The height of its bounding box.
- The position of the bottom of the bounding box along y axis.
- The position of the top of the bounding box along y axis.

The first measurement increases when a customer is reaching out before taking a product. The second measurement detects motion in products area. In fact, when a product is taken motion is detected where the product is missing. Furthermore, the measurements, related to the height and position of the bounding box, have meaningful variations as a person reaches out for products. The surface tends to increase as a person grasps a product, when products are big enough. These measurements fill the interaction context descriptor that possess 90 (6x15) dimensions, because we keep each measurement of the 15 last frames.

After running some tests, see results section 8, we decide to combine the local motion context and the interaction context description into one descriptor of 520 (430+90) dimensions, see figure 3.

7 Interaction Recognition

This section presents the behavior recognition process. Based on the behavior model, we detect the six states and build a Finite State Machine (FSM). Using video analysis and behavior description information, for each frame, the state of each object has to be identified among the six pre-defined states:

- **Enter** is detected when a person appears and is connected to an image boundary.
- **Exit** is detected when a previously tracked person is connected to an image boundary.
- **Interested** is detected when a person's contour connects a product area.
- **Stand by** is detected when a person is in the scene and not connected to a product area or an image boundary.
- **Inactive** is detected when the system loses track of a person. This event happens when a person has left the scene or is occluded by something in the scene, or another person.
- **Interaction** is detected using SVM [3], with a radial basis function kernel, on various descriptors presented above.

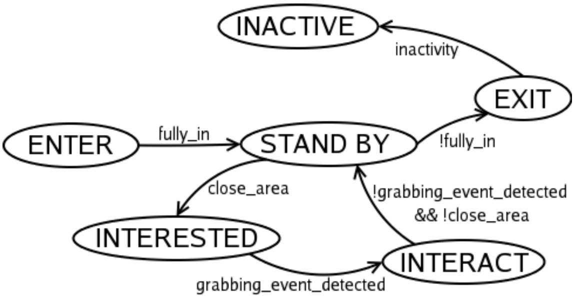


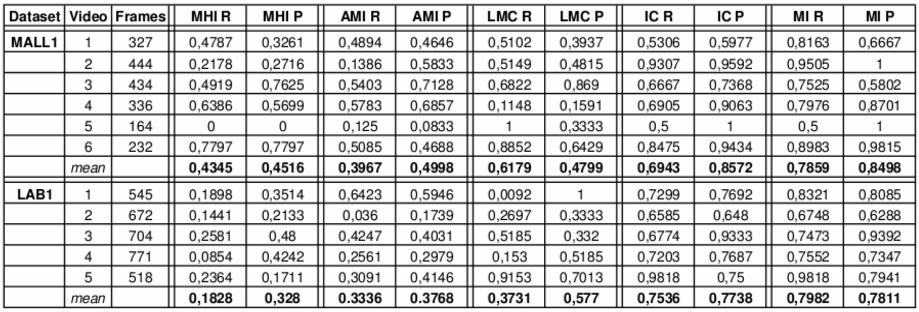
Fig. 4. Part of the Finite State Machine. All transitions are not written to make it clear.

Finite State Machine is used in order to organize and prioritize the six states [7]. 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. One FSM model the behavior of one person, see figure 4.

8 Results

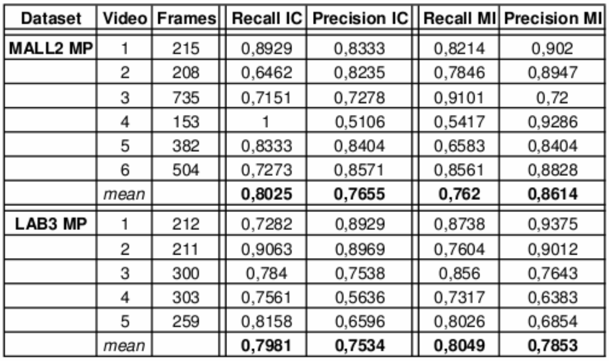
Datasets description: We use different datasets taken with the same camera, with 15 frames per second. A part of the datasets was taken in our laboratory (LAB1 and LAB3). The others were taken in a real shopping mall (MALL1 and MALL2). The two first datasets (LAB1 and MALL1) possess five and six sequences respectively and contains a lot of interactions with products. Two and four different people are shopping respectively, see figure 5. Products taken by people have different shapes, colors, and sizes in the video sequences. Furthermore, all products are identical in the heaps. LAB3 and MALL2 are two datasets where multiple people interact together. Two to four people interact simultaneously in the scene, see figure 5.

Table 1. Recall-Precision table for the Interact state for various descriptors on two datasets



Dataset	Video	Frames	MHI R	MHI P	AMI R	AMI P	LMC R	LMC P	IC R	IC P	MI R	MI P
MALL1	1	327	0,4787	0,3261	0,4894	0,4646	0,5102	0,3937	0,5306	0,5977	0,8163	0,6667
	2	444	0,2178	0,2716	0,1386	0,5833	0,5149	0,4815	0,9307	0,9592	0,9505	1
	3	434	0,4919	0,7625	0,5403	0,7128	0,6822	0,869	0,6667	0,7368	0,7525	0,5802
	4	336	0,6386	0,5699	0,5783	0,6857	0,1148	0,1591	0,6905	0,9063	0,7976	0,8701
	5	164	0	0	0,125	0,0833	1	0,3333	0,5	1	0,5	1
	6	232	0,7797	0,7797	0,5085	0,4688	0,8852	0,6429	0,8475	0,9434	0,8983	0,9815
	mean		**0,4345**	**0,4516**	**0,3967**	**0,4998**	**0,6179**	**0,4799**	**0,6943**	**0,8572**	**0,7859**	**0,8498**
LAB1	1	545	0,1898	0,3514	0,6423	0,5946	0,0092	1	0,7299	0,7692	0,8321	0,8085
	2	672	0,1441	0,2133	0,036	0,1739	0,2697	0,3333	0,6585	0,648	0,6748	0,6288
	3	704	0,2581	0,48	0,4247	0,4031	0,5185	0,332	0,6774	0,9333	0,7473	0,9392
	4	771	0,0854	0,4242	0,2561	0,2979	0,153	0,5185	0,7203	0,7687	0,7552	0,7347
	5	518	0,2364	0,1711	0,3091	0,4146	0,9153	0,7013	0,9818	0,75	0,9818	0,7941
	mean		**0,1828**	**0,328**	**0.3336**	**0.3768**	**0,3731**	**0,577**	**0,7536**	**0,7738**	**0,7982**	**0,7811**

Table 2. Recall-Precision Table for the Interact state using two descriptors on complex datasets



Dataset	Video	Frames	Recall IC	Precision IC	Recall MI	Precision MI
MALL2 MP	1	215	0,8929	0,8333	0,8214	0,902
	2	208	0,6462	0,8235	0,7846	0,8947
	3	735	0,7151	0,7278	0,9101	0,72
	4	153	1	0,5106	0,5417	0,9286
	5	382	0,8333	0,8404	0,6583	0,8404
	6	504	0,7273	0,8571	0,8561	0,8828
	mean		**0,8025**	**0,7655**	**0,762**	**0,8614**
LAB3 MP	1	212	0,7282	0,8929	0,8738	0,9375
	2	211	0,9063	0,8969	0,7604	0,9012
	3	300	0,784	0,7538	0,856	0,7643
	4	303	0,7561	0,5636	0,7317	0,6383
	5	259	0,8158	0,6596	0,8026	0,6854
	mean		**0,7981**	**0,7534**	**0,8049**	**0,7853**

Tests on datasets: In order to recognize product grabbing events, we use a cross validation process, see table 1 and 2. In other words, to recognize events on a video, we use all the other sequences of the dataset as training and then calculate recall and precision. It is interesting to note that using this process, we only use a few minutes of video as training with a few actors.

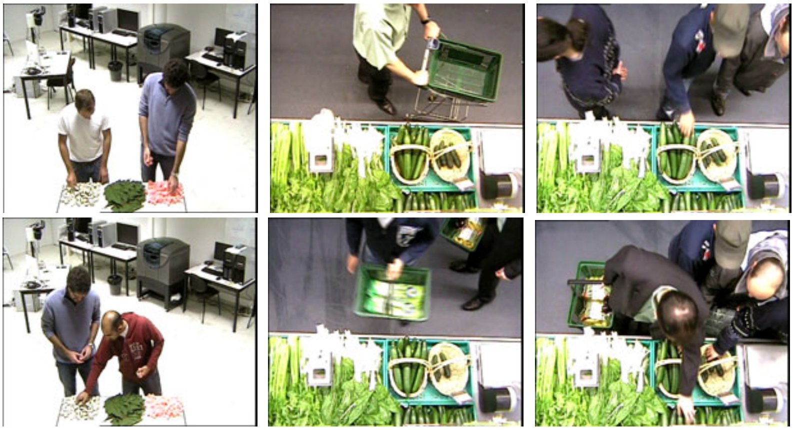


Fig. 5. Screenshots from LAB3 datasets on the first columns and MALL1 and MALL2 datasets on the right

After testing the four descriptors on various datasets, see table 1, we noticed that the appearance is not necessarily preserved from one sequence to another. Furthermore some videos show customers with shopping cart or basket that are detected as foreground. These detections modify completely the appearance of the persons.

Then we decided to combine the two descriptors offering the best results to test more datasets, see table 1. We compare result using only the Interaction context (IC) descriptor and combined local motion and interaction context (MI) descriptor. The MI performs as good as IC for precision, but offers better results for recall on the first datasets. On multiple people datasets, MI performs slightly better than IC on average. However, local motion description tends to be noisier, due to occlusions and a few object tracking mismatches. IC remains robust in these situations and the recognition rates in multi-person datasets are as good as in the first datasets, see table 2.

MALL performs better than LAB on recall and precision for the Interact state due to the position of the camera, located directly above the products and closer on MALL than on LAB, see table 2 and figure 5. We understand that the camera location is really important to maximize the accuracy of the system. A position close to the products performs better on Interact state recognition. 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.

Computation time: The application has to generate responses quickly as soon as a specific event is detected. The program is tested on a Pentium M, 1.73 Ghz with 500Mb of RAM. The application can analyze 6 to 10 frames per second for an image resolution of 704x576 or 640x480 respectively. Motion detection is the most computational expensive process.

9 Conclusion

This paper presents a novel type of application, using computer vision in the field of marketing, to improve interaction between customers and digital media. We evaluate various methods for behavior analysis and Interact context description outperforms the other methods. By combining Interact context and local motion context description, we improve these results. Interactions with products are well detected with a precision of 0.79 and a recall of 0.85.

As future work, the method can be generalized and tested on various complex scenes: products on vertical racks, complex products like clothes. It would also be interesting to look for other behaviors and scenarios that can be characterized and detected using this technique.

References

1. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. ICCV 2, 1395–1402 (2005)
2. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Trans. on Pattern Analysis and Machine Intel. 23, 257–267 (2001)
3. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV (2003)
5. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviours. IEEE Transac. on Syst., Man, and Cyb., 334–352 (2004)
6. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.S.: Action detection in complex scenes with spatial and temporal ambiguities. In: ICCV (2009)
7. Ikizler, N., Forsyth, D.: Searching video for complex activities with finite state models. In: CVPR (2007)
8. Kim, W., Lee, J., Kim, M., Oh, D., Kim, C.: Human action recognition using ordinal measure of accumulated motion. In: EURASIP JASP (2010)
9. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
10. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: 7th IJCAI, pp. 674–679 (1981)
11. PETS: Performance Evaluation of Tracking and Surveillance, `http://winterpets09.net/`
12. Poppe, R.: A survey on vision-based human action recognition. Im. & Vis. Comp. J. 28, 976–990 (2010)
13. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR (2008)
14. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR, vol. 3, pp. 32–36 (2004)
15. Sicre, R., Nicolas, H.: Shopping scenarios semantic analysis in videos. In: CBMI (2010)
16. Smeaton, A., Over, P., Kraaij, W.: Evaluation campaigns and TRECVid. In: ACM MIR (2006)
17. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
18. Turaga, P., Chellappa, R.: Machine recognition of human activities: a survey. IEEE Trans. on Circ. and Syst. for Video Tech. 18(11), 1473–1488 (2008)
19. Weinland, D., Ronfard, R., Boyer, E.: Free view-point action recognition using motion history volumes. CVIU (104), 249–257 (2006)
20. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surveys (2006)
21. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters 27(7) (2006)