Analyse de comportements dans les points de vente

R. Sicre[12] H. Nicolas[1]

[1] LaBRI (Laboratoire Bordelais de Recherche en Informatique)
Université de Bordeaux, 351 Cours de la libération, 33405 Talence Cedex - France

[2] MIRANE S.A.S.
16 rue du 8 mai 1945, 33150 Cenon - France

{sicre, nicolas}@labri.fr

Résumé

Cet article présente une nouvelle méthode permettant d'analyser, en temps réel, les comportements humains lors de l'acte d'achat. En particulier, nous cherchons à détecter l'intérêt d'une personne pour certains produits et des interactions telles qu'une personne saisissant des produits dans un point de vente.
Le système est basé sur un modèle de comportement. Le module d'analyse vidéo détecte le mouvement, suit les objets (personnes) dans la vidéo et décrit le mouvement local de ces objets. Les comportements définis dans le modèle sont ensuite reconnus. Enfin, nous testons le système sur des jeux de données réels.

Mots clefs

Vision par ordinateur, Analyse de comportement, Détection d'événements, Vidéosurveillance, Marketing.

1 Introduction

De plus en plus d'applications qui utilisent la vision par ordinateur sont développées dans divers domaines tels que la vidéosurveillance [8], surveillance de trafic routier [7], les jeux vidéos, le marketing, etc.
Dans le domaine du marketing, de plus en plus de points de vente s'équipent de systèmes d'affichage vidéo. Ces systèmes ont pour but de jouer des clips publicitaires les uns après les autres. Ils permettent un nouveau type de communication avec les consommateurs par le biais de clips d'animation, films, etc. Cependant l'impact de ces systèmes d'affichage est plus faible que l'on espérait. Ceci est dû au fait que les gens sont habitués à avoir de nombreux affichages publicitaires. C'est pourquoi le contenu et la localisation de ces systèmes ont besoin d'être étudiés. De nos jours, plusieurs logiciels sont disponibles sur le marché. En ce qui concerne la localisation des systèmes d'affichage, certains logiciels permettent de détecter et suivre les clients lors de leurs parcours dans le point de vente afin d'identifier leurs habitudes. D'autres logiciels calculent l'audience des systèmes d'affichage basée sur la détection des visages.

L'étude présentée dans cet article se situe dans le même contexte. Nous souhaitons adapter les clips affichés au comportement des personnes présentes afin d'améliorer l'impact des vidéos. Le système nécessite une phase d'analyse vidéo, suivie d'une phase de reconnaissance de comportement. Plus précisément, le système détecte, en temps réel, des personnes prenant des produits dans des zones connues. La détection d'un tel événement a pour résultat, par exemple, l'affichage d'un clip correspondant à l'objet saisi.
Apres une présentation des travaux précédents, l'article présente notre système. D'abord le modèle de comportement, puis l'analyse vidéo basée objet et la reconnaissance de comportement (voir figure 1). Nous présentons des résultats et proposons les travaux futurs en conclusion.

2 Travaux précédents

Les travaux précédents qui ont pour but de décrire les comportements humains, le font dans des contextes très variés [4] [12] [6]. Les êtres humains sont considérés comme des objets déformables. Le but de l'analyse de comportement est de reconnaitre des échantillons de mouvement afin d'en tirer des conclusions de haut niveau. Il y a beaucoup de problèmes à résoudre. Ceci est dû au fait que nous cherchons à mettre en correspondance des activités du monde réel et des données perçues par des systèmes de traitement des vidéos. Les buts sont ici de sélectionner des propriétés pertinentes générées par des méthodes d'analyse vidéo et de gérer leurs incertitudes.
L'analyse de comportement se fait généralement en deux étapes : description et reconnaissance des actions. La première étape vise à définir un modèle qui décrit chaque action pertinente dans notre contexte applicatif.
Ensuite il existe deux possibilités. D'abord, il y a une phase d'entrainement utilisant des données étiquetées pour ensuite reconnaitre les nouvelles données, basée sur cet entrainement. Les méthodes utilisées sont les modèles de Markov cachés, les réseaux de neurones, les Machines à vecteurs de support (SVM), etc.

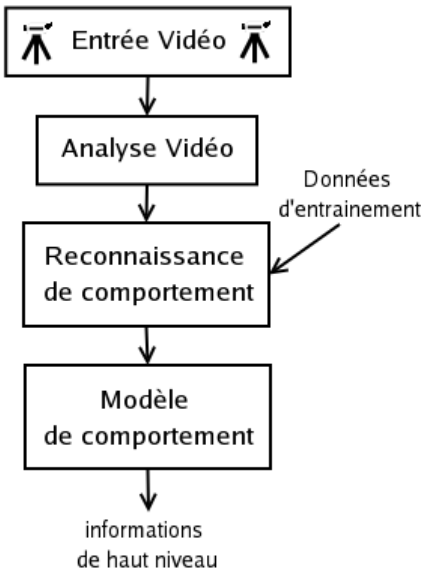

Figure 1 – *Diagramme fonctionnel du système*

Deuxièmement, un modèle logique est généré qui ne nécessite pas de phase d'entrainement. Cependant ces méthodes ne sont pas très flexibles et dépendent fortement de connaissances de la scène.

3 Modèle de comportement

Cette section présente le modèle qui définit le comportement des personnes lors de l'acte d'achat. Dans les points de vente les clients se déplacent, comparent les prix, prennent des produits, etc. Nous utilisons six états pour décrire le comportement des personnes présentes. La succession d'état décrit un scénario exécuté par une personne.

Enter : Une nouvelle personne entre dans la scène.
Exit : La personne sort de la scène.
Interested : La personne est proche des produits, probablement intéressée.
Interact : La personne interagit avec les produits, prend des produits.
Stand by : La personne est dans la scène, mais elle n'est proche ni des produits, ni d'une bordure de l'image. Cette personne peut se déplacer ou être arrêtée.
Inactive : La personne a quitté la scène.

4 Analyse vidéo basée objet

Afin de détecter les six états, nous avons besoin d'informations concernant chaque personne dans la scène. Nous voulons donc connaitre la position et le contour de chaque personne pour chaque image (ou frame) de la vidéo. C'est pourquoi, nous utilisons un procédé de détection de mouvement et de suivi d'objets. Nous supposons que les zones où se situent les produits sont connues.

Ensuite, la reconnaissance d'événements se fait grâce à la description et la classification du mouvement local d'une personne, de sa position relative aux zones de produits et de sa surface de recouvrement avec ces zones de produits.

4.1 Détection de mouvement - suivi d'objets

La détection de mouvement et le suivi d'objets déterminent le contour et la position de chaque objet (ou personne) en mouvement dans la scène pour chaque frame. La méthode utilisée se divise en deux phases : tout d'abord, la détection de mouvement identifie les régions en mouvement qui n'appartiennent pas à l'arrière-plan. Puis ces régions sont suivies le long de la vidéo. Les méthodes les plus rapides sont sélectionnées pour faire face aux contraintes de temps-réel de notre application.

La détection de mouvement utilise un modèle de l'arrière-plan basé pixel. Une mixture de gaussiennes est associée à chaque pixel afin de caractériser l'arrière-plan. Le modèle est mis à jour en ligne. Une distribution Gaussienne est mise en correspondance avec la valeur courante de chaque pixel. Si une Gaussienne appartient à l'arrière-plan, le pixel est classé de même. Sinon, il est considéré comme avant-plan (voir figure 2). Il est intéressant de noter que notre méthode permet aussi de tester si des pixels détectés comme appartenant à l'avant-plan correspondent en réalité à une ombre. Cette méthode nous permet d'effacer ces ombres et améliore grandement les résultats. Plus de détails concernant cette méthode se trouve dans [9] et [13]. Des filtres morphologiques sont appliqués sur le résultat de la détection afin d'effacer du bruit et d'améliorer la forme des régions détectées.

En pratique, un objet détecté peut être recouvert par plusieurs régions non connectées, car l'algorithme ne détecte pas certaines parties de la personne (voir figure 2 première colonne). C'est pourquoi, nous devons fusionner des régions. Les régions détectées sont représentées par leur boite englobante. Lorsque deux boites se superposent, les régions sont fusionnées. De plus, étant donné que nous suivons des personnes, nous supposons que les personnes sont situées dans des rectangles significativement plus hauts que larges, bien que le ratio dépende de chaque personne ainsi que de la position de la caméra. Par conséquent, deux régions situées dans le même axe vertical peuvent être considérées comme couvrant la même personne et ainsi être fusionnées. Dans certains cas, par exemple lorsqu'une seconde personne se trouve derrière la première, la fusion sera incorrecte. Cependant dans ce cas précis, la deuxième personne n'aura aucun intérêt dans notre contexte applicatif. Plus précisément, la fusion se produit lorsque le centre de gravité de la région supérieure se trouve entre les extrémités de la région inferieure. Chaque fusion est vérifiée par la phase de suivi.

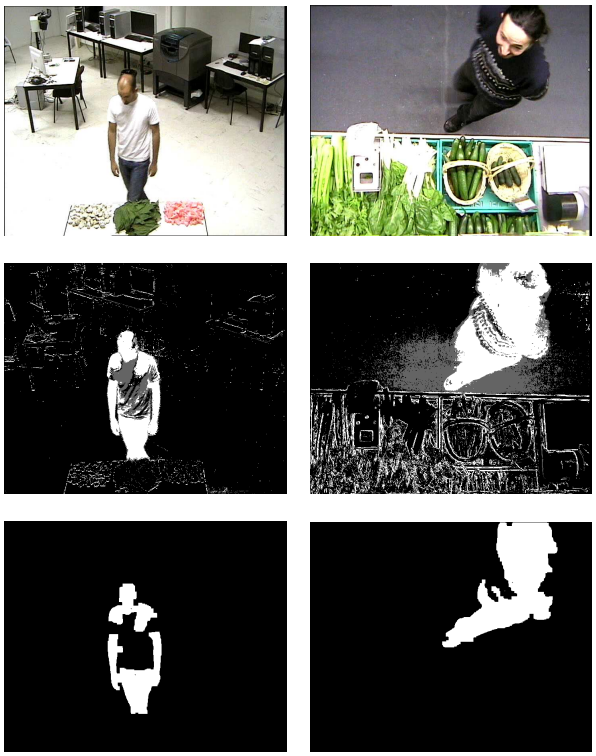

Figure 2 – *Détection de mouvement pour des vidéos dans les jeux MALL1 et LAB1. Les images de la premières ligne sont prises de la vidéo, la deuxième ligne correspond à la détection de mouvement avec detection des ombres (en gris) et la dernière est le résultat final après filtrage.*

Le suivi d'objets calcule d'abord pour chaque région un descripteur basé sur sa position, taille, surface, moment de couleur du premier et second ordre. Ces descripteurs sont ensuite mis en correspondance d'une image à l'autre, en utilisant un système de vote qui détermine les régions les plus similaires dans deux ensembles de régions. Finalement, les correspondances sont vérifiées grâce aux objets déjà suivis.

4.2 Description de la prise de produits

Cette section se focalise sur les interactions entre les personnes suivies et les zones de produits (personne prenant des produits). Lors de la prise d'un produit, une personne tend d'abord le bras, puis saisi un produit et finalement emmène le produit. Les différentes phases de cet événement de « prise de produit » correspondent à des mouvements locaux de cette personne, qui sont observables. Tout comme dans [3] et [10], nous utilisons un descripteur basé sur le mouvement local, mais aussi sur les interactions avec les zones de produits pour caractériser la prise de produit. Ce descripteur est utilisé dans la phase de reconnaissance de comportement (voir section 4) et est défini comme suit.

Le descripteur du mouvement local est calculé pour chaque frame et pour chaque personne suivie (voir figure 3). D'abord, la taille du masque de chaque personne est normalisée à une taille standard de 120x120 pixels, en gardant le rapport hauteur / largeur. Ensuite, le flow optique est calculé en utilisant l'algorithme de Lucas Kanade [5]. Cet algorithme nous retourne deux matrices contenants les valeurs des vecteurs de mouvement selon les axes x et y. Nous séparons les valeurs négatives des valeurs positives des deux matrices et obtenons 4 matrices avant d'appliquer un filtrage Gaussien sur chacune d'entre elles pour réduire le bruit. Une cinquième matrice est générée qui représente la silhouette de la personne suivie. Puis nous réduisons la dimensionnalité de ces matrices afin d'améliorer la vitesse du système. Chaque matrice est divisée en une grille 2x2. Nous intégrons les valeurs de chaque cellule de la grille selon un histogramme radial contenant 18 portions de 20 degrés chacune. Les matrices sont désormais représentées par un vecteur à 72 dimensions (2x2x18). Le descripteur complet contient lui 360 dimensions (5x72).
Pour prendre en compte les informations temporelles, nous utilisons 15 frames autour de l'image courante. Ces 15 frames sont divisées en trois groupes de 5 et représentent respectivement : le passé, le présent et le futur. Après avoir appliqué une Analyse en Composante Principale (PCA) sur les descripteurs de mouvement local de chaque frame du groupe, nous conservons les 50 premières composantes pour le groupe présent et les 10 premières pour les groupes passé et futur. Ce descripteur du contexte de mouvement local possède 70 dimensions (10+50+10), il est ajouté au descripteur précédent.

Le descripteur d'interaction utilise des informations provenant de la phase de suivi d'objets. Six valeurs sont calculées :

- La surface d'une personne recouvrant une zone de produits
- Un booléen qui est mis à 1 lorsque cette surface dépasse la taille théorique d'une main ou lorsque une personne est proche d'une zone de produits et que du mouvement est détecté dans cette zone.
- La surface totale recouverte par la personne dans l'image.
- La taille (hauteur) de la personne.
- La position du maximum de la personne selon l'axe y.
- La position du minimum de la personne selon l'axe y.

Les mesures concernant la taille et la position de la personne ont des variations intéressantes lorsqu'une personne saisit des produits. La surface recouverte par une personne dans l'image a tendance à augmenter lorsque celle-ci prend un produit, suivant la taille des produits. Ces mesures composent le descripteur d'interaction, qui possède 90 dimensions (6x15), car nous conservons les mesures des 15 frames.

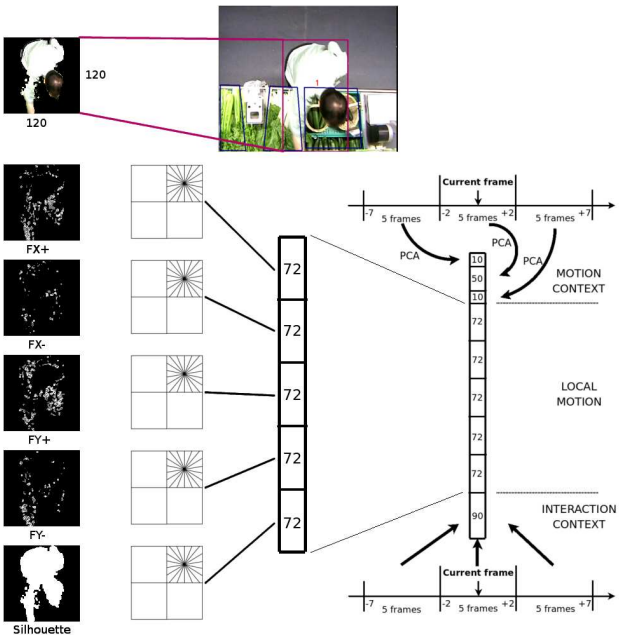

Figure 3 – *Diagramme représentant le descripteur*

5 Reconnaissance de comportement

Cette section présente le procédé de reconnaissance de comportement. Basés sur le modèle de comportement, nous déterminons les six états et créons une Machine à Etat Fini (MEF).

En utilisant les informations de l'analyse vidéo, nous devons définir pour chaque frame l'état dans lequel se trouve chaque objet (ou personne) suivi.

Enter est détecté lorsqu'une nouvelle personne est connectée à une bordure de l'image.

Exit est détecté lorsqu'une personne déjà suivie est connectée à une bordure de l'image.

Interested est détecté lorsque le contour d'une personne est connecté à une zone de produits.

Stand by est détecté lorsqu'une personne est dans la scène et qu'elle n'est ni connecté à une zone de produits ni à une bordure de l'image.

Inactive est détecté lorsque le système ne suit plus une personne, ou perd sa trace. Cet événement se produit lorsqu'une personne quitte la scène, ou est occulté par quelque chose dans la scène ou par une autre personne.

Interact est détecté en utilisant les SVM [2] sur le descripteur de mouvement local et d'interaction. Tout d'abord, il y a deux phases lors de la classification de données : l'entrainement et le test. Les données sont composées de plusieurs instances. Une instance est elle-même composée d'une valeur cible (target value) et de plusieurs attributs. Dans notre cas, la valeur cible est 0 ou 1, selon si une prise de produit a lieu ou non. Les attributs sont toutes les valeurs contenues dans le descripteur. Les SVM créent un modèle qui prédit les valeurs cibles à

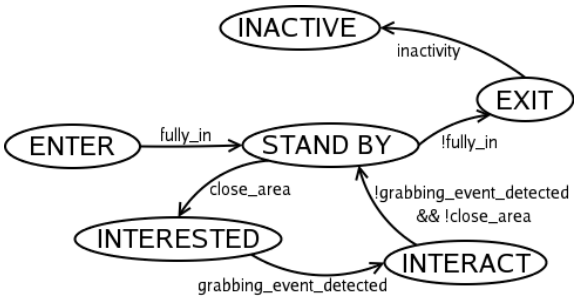

Figure 4 – *Portion de la machine à état fin. Toutes les transitions ne sont pas écrites pour une meilleure lecture.*

partir des attributs, en résolvant le problème d'optimisation suivant :

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

avec $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$ et $\xi_i \geq 0$

Le vecteur d'entrainement x_i est ajusté dans un espace à plus haute dimension par la fonction ϕ . Dans cet espace, les SVMs déterminent un hyperplan séparant les données et maximisant la variance. $\boldsymbol{C > 0}$ est le paramètre de pénalité (penalty parameter) de l'erreur. Le système utilise un noyau Gaussien (RBF).

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2), \gamma > 0$$

Bien que l'on puisse utiliser divers noyaux, nous choisissons le noyau RBF. En effet, il gère les relations non-linéaires entre les attributs et les valeurs cibles, contrairement aux noyaux linéaires. Les noyaux RBF ont aussi moins d'hyper-paramètres et de difficultés numériques que les noyaux polynomiaux ou sigmoid.

Une Machine à Etats Finis est créée afin d'organiser et de hiérarchiser les six états [1] [11]. La machine à état est synchrone et déterministe. La machine est synchrone, car elle itère à chaque nouvelle frame. Nous calculons le nouvel état en partant de l'état précédent et en testant toute les conditions de transition. Si une condition est satisfaite, le système se déplace dans le nouvel état correspondant. Sinon, le système reste dans le même état. La machine est déterministe, car pour chaque état, Il ne peut y avoir plus d'une transition valide pour chaque entrée possible. Une MEF modélise le comportement d'une personne. Nous sauvegardons le chemin de la personne à travers la MEF pour déterminer le scénario joué. La figure 4 montre un chemin possible à travers la MEF.

6 Résultats

6.1 Description des données

Nous utilisons plusieurs jeux de données pris avec la même caméra. Une partie des jeux de données ont été acquis dans notre laboratoire (LAB1, 2 et 3). Les autres ont été tournés dans un hypermarché (MALL1 et MALL2). Les jeux LAB1 et MALL1 possèdent cinq et six vidéos respectivement et contiennent beaucoup d'interactions avec les produits. Deux et quatre personnes différentes font leurs courses respectivement (voir figure2). LAB2 est un jeu dans lequel il n'y a pas d'interactions avec les produits et possède 3 acteurs. Les produits pris ont diverses tailles, couleurs, formes. De plus, tous les produits sont identiques dans les zones de produits. LAB3 et MALL2 sont deux jeux de données ou plusieurs personnes (deux à quatre) interagissent simultanément (voir figure 5).

6.2 Tests sur les états

Nous lançons des tests sur les données pour comprendre le comportement du système dans divers cas. Comme le système génère un état pour chaque personne et pour chaque frame de la vidéo, nous comparons ces résultats à la vérité terrain. Nous calculons ensuite le pourcentage d'états corrects (voir tableau 1). Le jeu de données LAB2, qui n'a pas d'interaction avec les produits obtient 89% et a un meilleur résultat que LAB1 avec 76%. Nous notons que l'état interact (prise de produits) est plus difficile à détecter que les autres états, pour les vidéos prises en laboratoire.

6.3 Tests sur la prise de produits

Pour reconnaître la prise de produits, nous utilisons un procédé de validation croisée (cross validation) (voir tableau 2). En particulier, pour reconnaître les événements d'une vidéo d'un jeu de données, nous utilisons toutes les autres vidéos de ce jeu comme entrainement et ensuite nous calculons la précision et le rappel.
Le jeu de données LAB1 offre de meilleurs pourcentages que MALL1. Cette différence est principalement due au bruit, qui est plus important dans l'hypermarché, ainsi qu'a des mouvements de caméra lors de la capture des vidéos. Cependant MALL1 offre de très bons taux de rappel et précision pour la détection de prise d'objet (voir tableau 2). Ceci est du à la position de la caméra, la caméra est directement au dessus des produits et plus proche que dans les vidéos tournées au Laboratoire. Les produits sont donc plus grands dans l'image et cela facilite leur détection. Dans le cadre d'une application concrète, un compromis devra être fait concernant la distance entre les zones de produits et la caméra. Une position proche des produits offrira de bons résultats pour la détection de prise d'objets. Cependant, un champ de vision trop réduit limitera les possibilités d'analyse du comportement des personnes.
Comme on le voit sur le tableau 2, nous avons comparé les résultats en utilisant deux descripteurs: le descripteur d'interaction seul (DI) et le descripteur de mouvement local avec le descripteur d'interaction (DMI). Le DMI a des performances similaires au DI en ce qui concerne la précision, mais offre un meilleur rappel. Pour les jeux de données avec plusieurs personnes, DMI a de meilleurs résultats en moyenne que DI. Cependant, le descripteur de mouvement local est plus bruité, cela est du à des erreurs de suivi.
La limite la plus importante du système réside dans le fait qu'il ne gère pas les phases d'occlusions. Cependant, la reconnaissance de prise de produits est robuste et offre de bons résultats pour les vidéos avec plusieurs personnes.
Pour conclure, nous pouvons faire quelques remarques générales concernant les produits. Ceux de petite taille sont logiquement plus difficiles à détecter lorsqu'ils sont pris. De plus, les produits clairs ont tendance à générer des fausses détections dues aux ombres des personnes lorsqu'elles sont proches. La méthode de détection des ombres atteint ses limites dans ce cas précis.

6.4 Temps d'exécution

Notre application doit générer une réaction en temps réel, dès qu'un événement est détecté. Le système est testé sur un ordinateur avec un Pentium 4, 3 Ghz et 1 Go de RAM. L'application analyse 6 à 10 images par secondes pour des résolutions de 704x576 ou 640x480 respectivement. La détection de mouvement est la phase la plus couteuse en temps de calcul.

7 Conclusion

Cet article présente un nouveau type d'application, utilisant la vision par ordinateur dans le domaine du marketing. Le système améliore les interactions entre les clients et les systèmes d'affichage, dans un point de vente. Celui-ci détecte, suit, et analyse les comportements des clients tels que les intérêts et les interactions avec les divers produits. Le système offre des résultats intéressants, 73% des frames sont correctement libellées pour des vidéos prises en environnement réel. Les interactions (prises d'objets) sont détectées avec une précision de 0.79 et un rappel de 0.85. Cette évaluation nous permet de comprendre le comportement du système pour augmenter son efficacité. Un prototype sera bientôt mis en place pour être testé sur une longue durée.
Notre méthode peut être améliorée avec un algorithme de gestion des occlusions. Il serait aussi intéressant de caractériser de nouveaux comportements et scénarios en utilisant la même technique.

Références

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

Dataset	Video	Frames	Correctness
MALL 1	1	327	70,03%
	2	444	74,77%
	3	434	66,13%
	4	336	70,83%
	5	164	76,22%
	6	232	79,74%
	mean		**72,95%**
LAB 1	1	545	85,87%
	2	672	74,40%
	3	704	76,28%
	4	771	60,57%
	5	518	92,66%
	mean		**76,13%**
LAB 2	1	476	87,61%
	2	342	82,46%
	3	143	91,61%
	4	303	95,71%
	mean		**89,35%**

Tableau 1 – *Tableau représentant le pourcentage d'état correctement détecté (correctness) pour chaque frame des vidéos.*

Figure 5 – *Quelques images provenant des jeux de données LAB3 et MALL2.*

Jeux	Video	Frames	Rappel I	Precision I	Rappel MI	Precision MI
MALL1	1	327	0,5306	0,5977	0,8163	0,6667
	2	444	0,9307	0,9592	0,9505	1
	3	434	0,6667	0,7368	0,7525	0,5802
	4	336	0,6905	0,9063	0,7976	0,8701
	5	164	0,5	1	0,5	1
	6	232	0,8475	0,9434	0,8983	0,9815
	mean		**0,6943**	**0,8572**	**0,7859**	**0,8498**
LAB1	1	545	0,7299	0,7692	0,8321	0,8085
	2	672	0,6585	0,648	0,6748	0,6288
	3	704	0,6774	0,9333	0,7473	0,9392
	4	771	0,7203	0,7687	0,7552	0,7347
	5	518	0,9818	0,75	0,9818	0,7941
	mean		**0,7536**	**0,7738**	**0,7982**	**0,7811**
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**

Tableau 2 – *Tableau représentant le rappel et la précision pour la détection de prise de produits. Deux descripteurs sont testés sur plusieurs vidéos : descripteur d'interactions (I) et le descripteur de mouvement local et d'interaction (MI).*

[2] C. Chang and C. Lin, "LIBSVM: a library for support vector machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[3] A. A. Efros, A.C. Berg, G. Mori, J. Malik, "Recognizing action at a distance", *Int. Conf. on Computer Vision*, 2003.

[4] 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.

[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", *in Proc. 7th IJCAI*, pp.674–679, 1981.

[6] T. Moeslund, A. Hilton, V. Kruger, "A survey of advances in vision-based human motion capture and analysis", *Computer vision and image understanding*, pp 90-126, 2006.

[7] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1114–1127, 2008.

[8] PETS: Performance Evaluation of Tracking and Surveillance, http://winterpets09.net/

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

[10] D. Tran, A. Sorokin, "Human activity recognition with metric learning", *Euro. Conf. on Computer Vision*, 2008.

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

[12] A. Yilmaz, O. Javed, M. Shah, "Object tracking: a survey", *ACM Computing Surveys*, 2006.

[13] Z. Zivkovic, F. van der Heijden "Efficient adaptive density estimation per image pixel for the task of background subtraction" *Pattern Recognition Letters*, vol. 27, no. 7, pages 773-780, 2006.