Using HMMs for Discriminating Mobile from Static Objects in a 3D Occupancy Grid

Amandine Dubois, Abdallah Dib, François Charpillet

Using HMMs for Discriminating Mobile from Static Objects in a 3D Occupancy Grid

Amandine Dubois*, Abdallah Dib*, François Charpillet*,
*INRIA Nancy - Grand Est and LORIA
615 rue du Jardin Botanique
54600 Villers-lès-Nancy
Email: firstname.lastname@inria.fr

***Abstract*—This work is related to the development of a markerless system allowing the tracking of elderly people at home. Microsoft Kinect is a low cost 3D camera adapted to the tracking of human movements. We propose a method for making the fusion of the information provided by several Kinects. The observed space is tesselated into cells forming a 3D occupancy grid. We calculate a probability of occupation for each cell of the grid. From this probability we distinguish whether the cells are occupied or not by a static object (wall) or a mobile object (chair, human being). This categorization is realized in real-time using a simple three states HMM. The proposed method for discriminating between mobile and static objects in a room is the main contribution of this paper. The use of HMMs allows to deal with an aliasing problem since mobile objects result in the same observation as static objects. The approach is evaluated in simulation and in a real environment showing an efficient real-time discrimination between cells occupied by mobile objects and cells occupied by static objects.**

***Keywords*-occupancy grid; Hidden Markov Model, mobile object tracking, Depth image;**

I. Introduction

The ultimate objective of our project is to allow elderly people to live at home while they lose their autonomy. One of the main preoccupations as regard to the safety of this category of person, is to prevent falls. Several approaches have addressed this issue. Dib *et al.* [1] used a dynamic bayesian network (DBN) and a factored particle filtering algorithm for building a markeless human motion capture system estimating the 3D positions of the body joints over time. In this work several observation functions were build from the images provided by RGB cameras. The bayesian approach allowed to make the fusion of the different observation functions. Having a rich observation of the system allows to reduce the number of particles in the particle filter. Other researchers have successfully used 3D cameras for home monitoring of elderly. Jansen *et al.* [2] used a 3D camera in order to classify the pose of a person among an *a priori* set of characteristic poses: "standing", "sitting" or "lying down".

In this paper, we propose a system based on range cameras such as Kinect for tracking human movements. The Kinect was preferred to other camera systems because it has the particularity of being a low cost device sending back a color image and a depth image.

This paper deals with two issues regarding the use of several Kinects for tracking human movements. The first one is the fusion of the information provided by the different cameras. The second one is the discrimination between mobile objects and background. We will show that both problems can be addressed using a three states HMM.

From the depth image we create a spatial representation called an occupancy grid. In this representation, the space is divided into cells of a few centimeters with a probabilistic occupation state. In our approach a 3D occupancy grid is used for making the fusion of the information provided by several 3D cameras.

The second problem for tracking persons is to succeed at discriminating mobile objects from the background. Our approach is based on an extension of occupancy grids [4] using hidden Markov models in place of bayesian filtering such that each voxel of the grid is determined by a three state model (the voxel belongs to the background, the voxel belongs to a mobile object, the voxel is non occupied). Our work is related to others. Among them, let us quote Yapo *et al.* [3] who proposed a method to detect 3D objects using LIDAR sensors. Their approach is also based on the concept of occupancy grids. From a probabilistic representation they determine if the voxels are free, occupied or hidden.

This paper is organized as follows. In section II, the background on occupancy grid is presented. Then, in section III the method we propose is described. Subsection A is dedicated to the data fusion of several 3D sensors. Then subsection B describes the observation function that we use in the HMM. Subsection C examines how inference are made in order to estimate state probabilities. Finally in subsection D, we present experimental results obtained with our method.

II. Background on occupancy grids

Occupancy grids such as defined in the article of Alberto Elfes [4] consist in dividing into cells a 2D or in our case a 3D space. The grid provides a representation of the environment. For each cell we estimate its state, which can be either occupied or empty, from a probability of occupation. For the sake of simplicity, each cell is estimated independently of its nearby cells. A sensor model is used to calculate the probability of occupation. This model, denoted as $P(r|z)$ with z the value the sensor should read and r the sensor reading, has the form of a gaussian distribution. In case of a range sensor, the probability of occupation $P(s(x)|r)(x)$ (with $s(x)$ a cell of the occupancy grid and z the distance the sensor should read if the cell is occupied and not occulted by an other object, and r the sensor reading) is represented by the Figure 1. This probability is updated using the Bayes rule. The estimation of the state of a cell i (positioned at $x = C_i$) can be calculated as:

$$p_{t+1} = \frac{P(r_{t+1}|s(C_i) = occ)p_t}{P(\{r\}_{t+1})}$$

where

$$p_{t+1} = P(s(C_i) = occ|\{r\}_{t+1})$$

and

$$P(\{r\}_{t+1}) = \sum_{s(C_i)} P(r_{t+1}|s(C_i))P(s(C_i)|\{r\}_t).$$

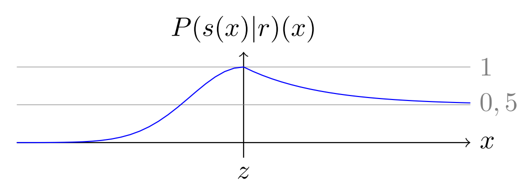


Figure 1. Occupancy Grid.

We can consider the occupancy grid model as a two states HMM with no transitions between states as shown in Figure 2.

Figure 2. Representation of the occupancy grid model (W: occupied; E: empty).

III. Method

A. Kinect fusion

Kinects must be calibrated to a global coordinate system in order to merge several Kinects and to be able to update the occupancy grid for each Kinect. Our common coordinate system being the ground level, we need to calculate the transformation matrix between the ground and each camera. Usual method for camera calibration like epipolar geometry [5] or chessboard calibration can be used to calibrate RGB cameras with respect to ground.

Let $K_{RgbGround}$ be the transformation matrix between the RGB camera and a common coordinate system. The transformation matrix $K_{RgbDepth}$ between the depth camera and the RGB camera of each Kinect being known, the transformation $K_{GroundDepth}$ between the depth camera and the common coordinate system can be obtained easily with the following equation:

$$K_{GroundDepth} = K^{-1}_{RgbGround} \times K_{RgbDepth}$$

Once calibrated, each 3D point in the Kinect coordinate system can be easily projected to the ground by multiplying by the inverse of $K_{GroundDepth}$. Hence the 3D points of different Kinects can be projected to the ground coordinate system as illustrated in the Figure 3.

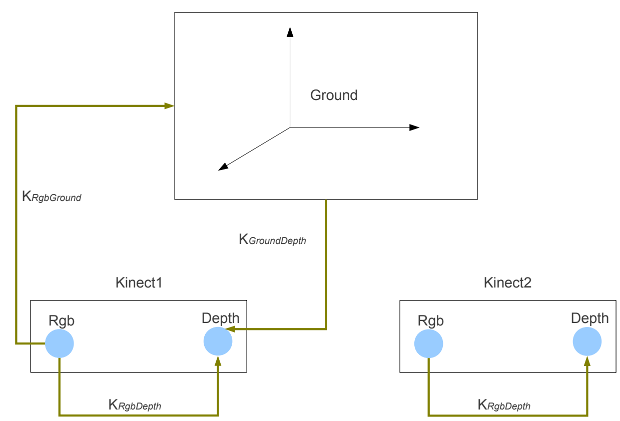


Figure 3. Kinects ground calibration.

The occupancy grid is defined in the common coordinate system which allows to project the cells of the grid to the different camera coordinate systems and define an observation function for each Kinect.

B. Observation function

Each voxel C_i is represented by its center of mass, defined by coordinates (x,y,z). We can obtain at which distance is located the voxel from the camera by projecting the voxel to the camera coordinate system using the camera transformation matrix $K_{GroundDepth}$. We denote as l this distance. The distance l of the voxel is compared to the depth, denoted as d, of the corresponding pixel provided by the Kinect camera. The observation r (see the section II) takes as value the error of distance (ε) between d and l calculated as $\varepsilon = d - l$. An observation function is built to evaluate the probability of occupation of the cell from the depth image $P(r|C_i) = f(\varepsilon)$. $f(\varepsilon)$ is represented in Figure 4.

This function represents the following three cases:

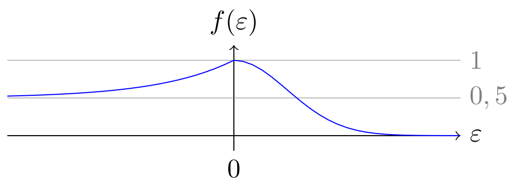


Figure 4. Representation of probability occupation.

- for $\varepsilon > 0$: the state of the voxel is empty because the first object is located at a distance superior to the one of the voxel,
- for $\varepsilon = 0$: the state of the voxel is occupied by the first object visible by the camera,
- for $\varepsilon < 0$: the state of the voxel is unknown because it is masked by an object closer to the camera.

Assuming that the information provided by the different camera is conditionally independent, we can multiply the different observation functions:

$$P(r_1, ..., r_N|C_i) = \prod_{j=1}^{N} f(\varepsilon_j)$$

where N is the number of cameras.

C. Probability of occupation

In the classical occupancy grid method the state "empty" or "occupied" is calculated from the probability of occupation $P(r|C_i)$. For each cell, we use a three states HMM allowing to represent its dynamic and to determine its state. The three states are:

- the state "W" meaning that the cell is occupied and has always been occupied;
- the state "O" meaning that the cell is occupied but has already been empty at least once;
- the state "E" meaning that the cell is not occupied.

The representation of this HMM is shown in Figure 5. with

$$\begin{aligned} \alpha &= 0.01 \\ \beta &= 0.1 \\ \gamma &= 0.4. \end{aligned}$$

The probability to be in one of the three states is calculated with the Forward procedure [6]. We denote

$$\begin{aligned} pw_t = P((C_i)_t &= A|(r_t, r_{t-1}...r_0)) \\ po_t = P((C_i)_t &= S|(r_t, r_{t-1}...r_0)) \\ pe_t = P((C_i)_t &= E|(r_t, r_{t-1}...r_0)) \end{aligned}$$

$$\begin{aligned} P(r|C_i = pw) &= f(\varepsilon) \\ P(r|C_i = po) &= f(\varepsilon) \\ P(r|C_i = pe) &= 1 - f(\varepsilon). \end{aligned}$$

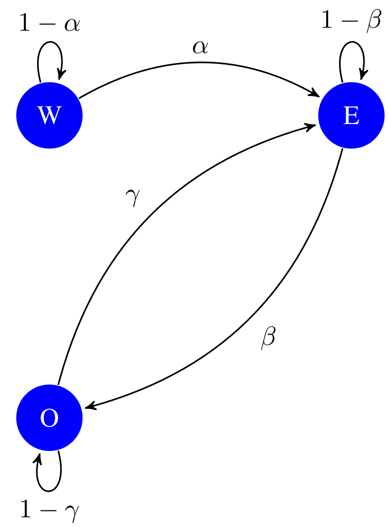


Figure 5. A similar HMM is used to model the evolution of each cell.

The reckoning of the probabilities used for each state is the following:

$$\begin{aligned} g &= [pw_{t-1} * b] * f(\varepsilon) \\ h &= [po_{t-1} * c + pe_{t-1} * e] * f(\varepsilon) \\ i &= [pe_{t-1} * f + pw_{t-1} * a + po_{t-1} * d] * (1 - f(\varepsilon)) \end{aligned}$$

$$\begin{aligned} pw_t &= \frac{g}{g+h+i} \\ pol_t &= \frac{h}{g+h+i} \\ pe_t &= \frac{i}{g+h+i} \end{aligned}$$

with the following initialization (see section IV-C):

$$pw_0 = 0.5$$

$$po_0 = 0.0$$

$$pe_0 = 0.5.$$

D. Categorization of cells

We want to distinguish mobile objects (chair, being humans) from static objects (walls). We define a cell containing a mobile object as being a occupied cell that has previously been empty. Whereas cells containing a static object are cells that are occupied and that have never been empty. In other words we can write:

- for dynamics objects: $C_i = O$
- for statics objects: $C_i = W$
- for cells occupied by object: $C_i = O \vee W$

Cells are categorized by choosing the maximum *a posteriori* (MAP), that is to say the most likely state of the corresponding HMM.

Number of cameras	1	2	3	4	5	6
FPS	30	27	22	17	14	11

Table I
EXECUTION SPEED OF THE ALGORITHM, MEASURED IN FRAME PER SECOND (FPS) WITH DIFFERENT NUMBER OF CAMERA AND 1 147 500 CELLS.

IV. RESULTS

A. Implementation

Since the occupancy calculation is local for each cell in the grid, the algorithm can be easily parallelized. A GPU implementation was made for maintaining and updating the grid using OpenGL pixel shader stage.

In order to test the performances of the algorithm in term of execution speed, we calculate the number of frame per seconds (FPS) which we obtain in various situations that we present in the Table I.

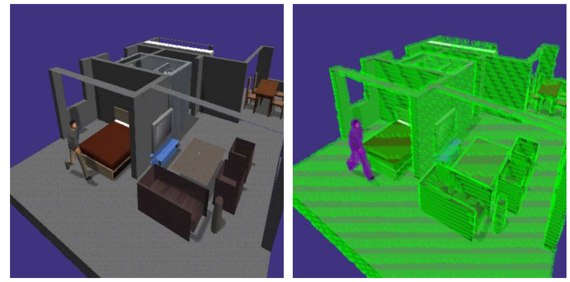
These results were obtained using a 2.53GHz quad core laptop with a GeForce GT 330M GPU.

The results with more than two cameras were obtained by simulating the supplementary cameras. The simulation consists in repeating several times the operations realized for one camera. The Kinect can return up to 30 images per seconds according to its manufacturer.

B. Simulation

This section describes our method for evaluating the sensitivity and the specificity of the system in a simulated environment. The sensitivity is the capacity to detect mobile objects when they are present and the specificity is the capacity of the system to detect the absence of mobile objects when there is no mobile object.

Figure 6 shows the output of the simulator and the result of the system classification. In order to perform the evaluation, the output of the system should be compared to a reference image pixel by pixel. Since it is impossible to evaluate the system in real conditions due to the fact that we need to index real images, we opted for an evaluation in a simulated environment. We have recorded a simulated human activity in a virtual scene and used these images as a reference. We simulate a Kinect by generating depth and RGB images from the virtual scene. In addition, a reference image that index each pixel in the scene as static or mobile object is also generated. RGB and depth images are supplied to our system to perform the classification. Finally we compare the output of the algorithm to the reference image. Results show a sensitivity of 87.14% and a specificity of 98.52% for a total of 430 frames (73.8M pixels). In the reference images only 2% of the pixels corresponded to the moving person whereas the other 98% were static objects or background. Table II shows the number of pixels for static and mobile objects obtained from the simulation and detected by our system.



(a) The simulation. (b) The apartment simulated with distinction between static and mobile objects.

Figure 6. The apartment simulation.

		reference	
		pixels: mobile	pixels: static
detected	pixels: mobile	1 294 005	1 068 236
	pixels: static	190 958	71 278 387

Table II
NUMBER OF PIXELS IN EACH CATEGORY.

C. Initial conditions

This part of the discussion concerns the initial probability distribution $\pi = (pw_0, po_0, pe_0)$. Having no prior information about the scene, we suppose that all cells can be initially empty or occupied with the same probability. The meaning of the state O of the HMM being that the cell is occupied but has been previously seen empty, we consider that its initial probability is ($po_0 = 0$). Therefore the initial probability of state W and E are fixed at 0.5 ($pw_0 = pe_0 = 0.5$).

With this initial configuration, the model behaves very well at recognizing cells occupied by static objects. Cells occupied by mobile objects are identified as soon as the object starts to move.

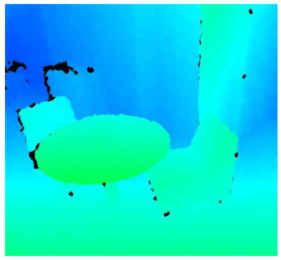
In order to evaluate the sensibility of the model to the initial conditions, we have tested the HMM with different initial probabilities. We tried to equalize the initial probability of each state ($pw_0 = po_0 = pe_0 = 1/3$). The equality of the probabilities didn't really affect the results. After a short period of confusion between mobile and static objects the model converges to the same results as with the previous initial conditions. More generally we can see that the model is robust to initial conditions.

This robustness shows that the categorization of the cells relies on the transition matrix of the model. Cells containing static object are identified due to the difference between transitions $P(W_t|W_{t-1}) = 1-\alpha$ and $P(O_t|O_{t-1}) = 1-\gamma$, more precisely because $\gamma > \alpha$.

D. Behavior in realistic conditions

We have tested our algorithm in an experimental apartment. Results presented here are qualitative. In Figure 7(a) and 7(b) we see the RGB and depth images. The image sent

(a) RGB image Kinect.



(b) Depth image Kinect.

(c) 3D reconstruction of the scene using depth and RGB images.

Figure 7. Image of Kinect camera.

back by the Kinect is illustrated by Figure 7(c). The black space corresponds to a badly reconstructed zone.

We have tested the algorithm with one camera and one person walking in front of the camera. As illustrated by Figure 8, walls, furnitures and the ground are correctly detected as static objects represented by green color and the person as a mobile object represented by blue color. We can see that the feet of the person in figure are detected as a static object, it's due to the size of the voxels and the uncertainty of the observation. The voxels have a length of 6 cm. The feet of the person are integrated in voxels representing the ground. We can notice that there is very limited noise on the background where a few badly positioned blue cubes remain. Moreover the tracking of mobile objects is fast enough to distinguish visually the members (leg, arm) of the person as it can be seen on Figure 8. A space without color is present above in the left of this figure. This is due to the size of the grid which is limited here to the perception range of the Kinect. The Kinect can reconstruct the depth of the scene up to 4 meters.

The obstacles in a room don't disturb the discrimination between mobile and static objects as shown on Figure 9.

When we move a furniture, this furniture, previously detected as a static object, is recognized as a mobile object. Figure 10 shows a chair becoming a mobile object. This result is allowed by the transition ($W \rightarrow E$) which models the fact that a furniture can be moved resulting in new empty space.

After a certain amount of time, it could be interesting to consider a furniture that has been moved as a new static

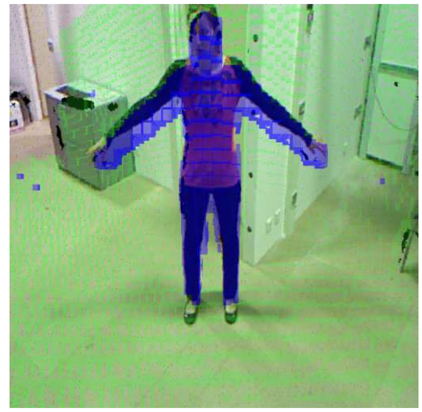


Figure 8. Green color: static objects (walls, furnitures, the ground). Blue color: mobile object (person).

Figure 9. Sitting person in a environment with obstacles.

object. This can be realized simply by adding a link ($O \rightarrow W$) with a small probability γ_2 to the transition matrix as illustrated by Figure 11.

We have also tested the algorithm in a situation where several persons are walking in the field of view of the camera. This test showed that all persons were correctly detected as mobile objects as shown by Figure 12.

(a) The chair is considered as a static object.

(b) The chair has been moved and is considered as a mobile object.

Figure 10. Chair becoming a mobile object.

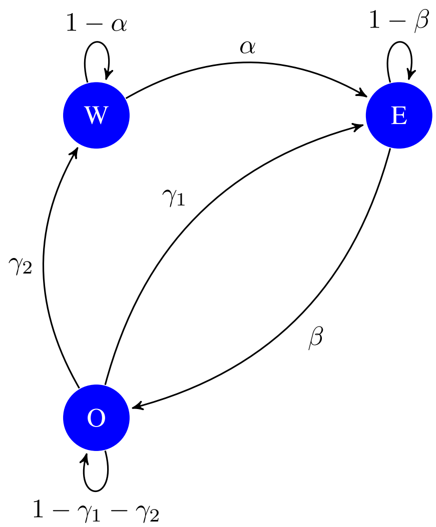


Figure 11. Adding a link ($O \rightarrow W$) to the HMM so as to allow for a moved furniture to become a static object.

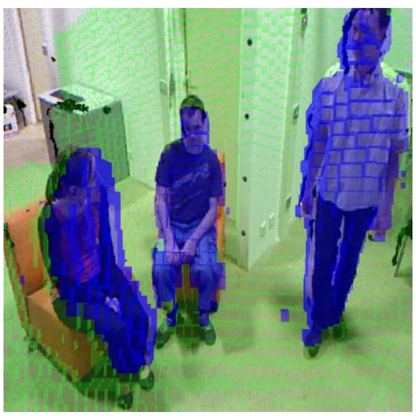
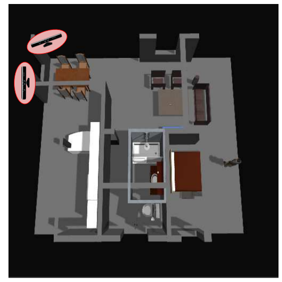


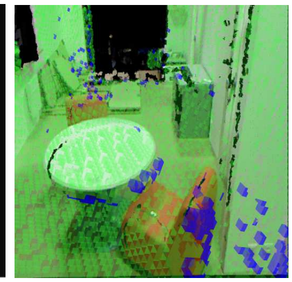
Figure 12. Several persons are in the room.

The experiment was then realized with two cameras placed as illustrated on Figure 13(a). We can see that the fusion of several cameras allows to discover more space while decreasing the noise around static objects as illustrated on Figure 13.

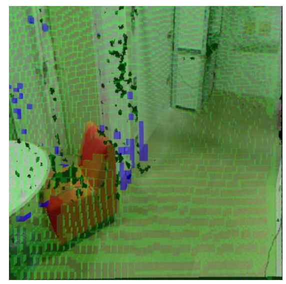
To finish we have tested the algorithm not in an experimental apartment but in a true apartment. One of the differences is on the level of lighting. The experimental apartment is located in a larger room with wall painted in black and not having windows. Thus lighting comes primarily from the artificial light. We wanted to test the algorithm in a more natural scene. We can see in Figure 14 that there are less noise compared to the experimental apartment. But we have noticed that when there is too much sun light on a white surface, the Kinect badly reconstructs the zone which is represented by black color in the lower right corner of Figure 14(a).



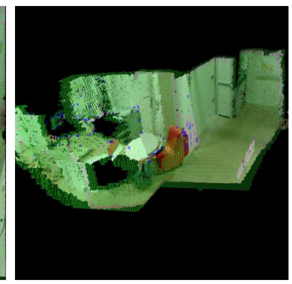
(a) Position of the two cameras in the apartment.



(b) View of one of the cameras.

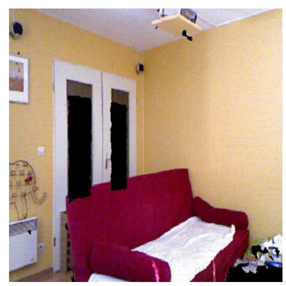


(c) View of the other cameras.

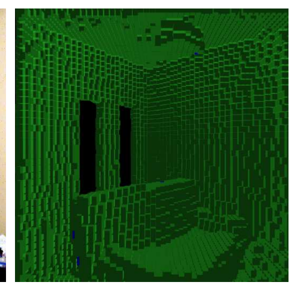


(d) Fusion of the two cameras.

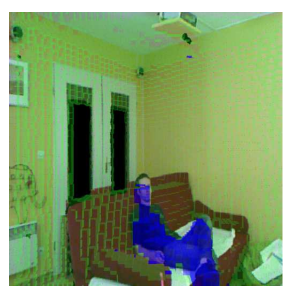
Figure 13. Test with two cameras.



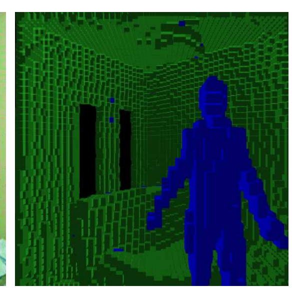
(a) View of camera Kinect.



(b) All the objects are detected as static (view without texture).



(c) A person is detected as a mobile object.



(d) A person is detected as a mobile object (view without texture).

Figure 14. The use of the algorithm in a real apartment.

V. Conclusion

We presented a markerless system using Kinect cameras in the aim of tracking of elderly people at home. First we proposed a system to merge several cameras by using 3D occupancy grid. We divided the scene into cells following the method of occupancy grid. Secondly compared to previous work on occupancy grid we proposed a method to allow the tracking of mobile objects. This method is based on a three states HMM: cell is empty, cell has always been occupied (static objects) and cell is occupied but has already been empty (mobile objects). This three state HMM is a simple yet elegant solution for solving a state aliasing problem (the observation for a static object is the same as the observation for a mobile object). Since each cell is updated independently one of the other, the process can be easily parallelized and implemented in a GPU allowing real-time (30 FPS) tracking with 2 cameras on a 1M cell grid.

Results in simulation allowed us to measure the quality of classification performed by the system in terms of sensitivity and specificity.

Results on real images concerning the detection of cells occupied by mobile objects are visually satisfying.

Several problems have to be treated in continuation of this work like tracking a person among a group, detecting the activity of this person. The purpose is to learn the habits of a person for thus detecting when an unusual behavior occurs. Another point to be treated will be to follow the evolution of the gait of the person so as to prevent falls. According to the study of Auvinet *et al.* [7] the irregularity of the steps is regarded as a relevant variable for the prediction of falls. Thus one continuation of this work will be to extract the length of the person steps. One of the problems caused by the use of several cameras in each room of an apartment is that it is an intrusive method. An other research direction would be to install a Kinect on a robot. The new difficulties would then be to update the occupancy grid and to determine the position of the camera at the same time.

VI. Acknowledgement

This work has been partly funded by Region Lorraine. The authors thanks Cedric Rose for his valuable comments on this work.

References

[1] A. Dib, C. Rose, and F. Charpillet, "Bayesian 3D human motion capture using factored particle filtering," in *Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI'10)*, 2010.

[2] B. Jansen, F. Temmermans, and R. Deklerck, "3D human pose recognition for home monitoring of elderly," in *Proceedings of the 29th IEEE EMBS annual international conference*, August 2007.

[3] T. Yapo, C. Steward, and R. Radke, "A probabilistic representation of LiDAR range data forefficient 3D object detection," in *Proceedings of the S3D (Search in 3D) Workshop, in conjunction with IEEE CVPR*, June 2008.

[4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, June 1989.

[5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge University Press, 2003.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[7] B. Auvinet, G. Berrut, C. Touzard, and L. Moutel, "Gait abnormalities in elderly fallers," *Journal of aging and physical activity*, 2003.