Tracking Mobile Objects with Several Kinects using HMMs and Component Labelling

Amandine Dubois, François Charpillet

Tracking Mobile Objects with Several Kinect using HMMs and Component Labelling

Amandine Dubois[1,2] and François Charpillet[2,1]

Abstract— **This paper proposes a markerless system whose purpose is to detect falls of elderly people at home. To track human movements we use Microsoft Kinect camera which allows to acquire at the same time a RGB image and a depth image. One contribution of our work is to propose a method for fusioning the information provided by several Kinects based on an occupancy grid. The observed space is tesselated into cells forming a 3D occupancy grid. We calculate a probability of occupation for each cell of the grid independently of its nearby cells. From this probability we distinguish whether the cells are occupied by a static object (wall) or by a mobile object (chair, human being) or whether the cells are empty. This categorization is realized in real-time using a simple three states HMM. The use of HMMs allows to deal with an aliasing problem since mobile objects result in the same observation as static objects. This paper also deals with the issue regarding the presence of several people in the field of view of camera by using the component labelling method. The approach is evaluated in simulation and in a real environment showing an efficient real-time discrimination between cells occupied by different mobile objects and cells occupied by static objects.**

I. Introduction

During the last fifty years, the many progresses of medicine as well as the improvement of the quality of life have resulted in a longer life expectancy. The increase of the number of elderly people is a matter of public health because although elderly people can age in good health, old age also causes embrittlements in particular on the physical plan which can result in a loss of autonomy. If we look at elderly people (people of more than 65 years old), we realize that one of the main preoccupations at this age is fall. Indeed by looking at the figures of the INPES [1] we notice that one elderly people out of three falls in the year. Our work is related to fall prevention and detection.

Many systems exist to detect falls. One of the categories consists in systems with sensors, that the person wears on her. These sensors are either accelerometer, gyroscopes or goniometers. These various sensors can be integrated in devices detecting the fall automatically, as shown in article of Bourke *et al.* [2]. There exists also systems made of an alarm button, in this case it's the person who must press herself on a button to alert after the fall. Another category of fall detection devices are the systems using 2D or 3D cameras. For example Jansen *et al.* [3] 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". For working on detection falls, we interessed us in this last category of devices.

We chose a RGB depth camera, more precisely the Kinect camera. One of the disadvantages of the Kinect camera is its rather reduced field of view. The sensor can maintain tracking approximately 0.7–6 m. However we want to detect falls in parts of an apartment often larger than what the Kinect camera is able to see. Thus to cover a room entirely we should integrate several cameras in the same room.

This article is focused on the problem of human movements tracking with several Kinect. For this, three problems need to be solved. 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. The last problem consists in gathering the mobile elements belonging to the same object. This stage makes it possible to identify several people as distinct in the same scene. In earlier work [4], we presented a method to detect pixels belonging to a mobile object adressing to the two first problems of human movements tracking. This paper entends previous work by adressing the third problem. We identify pixels belonging to the same object.

For making the fusion of the information provided by several 3D cameras, our method is to use a 3D occupancy grid. 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.

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 [5] using hidden Markov models 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 not occupied). Our work is related to others. Among them, let us quote Yapo *et al.* [6] 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.

The third problem of identification of the mobile objects is carried out thanks to the component labelling method which consist in gathering the voxels belonging to the same mobile object.

This paper is organized as follows. Section II is dedicated to the data fusion of several Kinect cameras. Then, section III describes the method for categorizing the voxels (cells) occupation state. Section IV explains the method to identify

[1]Université de Lorraine
[2]INRIA
LORIA – Campus scientifique, BP 239, 54506 Vandœuvre-lès-Nancy CEDEX

the different mobile objects. Finally in section V, we present experimental results obtained with our method.

II. Kinect fusion

We use an occupancy grid as a common coordinate system so as to fusion information from several Kinect cameras.

A. Occupancy grids

Occupancy grids such as defined in the article of Alberto Elfes [5] 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 C_i (or voxels) we estimate its state, which can be either occupied or empty, from a probability of occupation $P(C_i|r)$. The probability of occupation is the probability of cell C_i to be in state occupied given sensor reading r. For the sake of simplicity, each cell is estimated independently of its nearby cells.

B. Calibration of Kinect

The fusion information of several Kinect consists in determining the position and orientation of the different Kinect compared to the common coordinate system (the grid). For that we calibrate the Kinect cameras. The aim of camera calibration is to determine the transformation matrix (represented in Figure 1) between the RGB camera of each Kinect (K_{RgbRgb}), between RGB camera and the grid ($K_{RgbGrid}$), between RGB and depth camera of the same Kinect ($K_{RgbDepth}$) and between grid and depth camera ($K_{GridDepth}$). So with all these transformation matrix, the cells of the grid can be projected to the different camera coordinate systems (Kinect) and we will be able to define an observation function for each Kinect presented as in section III-B.

The transformation matrix K_{RgbRgb} and $K_{RgbGrid}$ can be determined with the method of epipolar geometry [7] or chessboard calibration.

The transformation matrix $K_{RgbDepth}$ is known by manufacturer of Kinect (provided by openNI).

The transformation matrix $K_{GridDepth}$ is deduced from previous transformations with the following equation :

$$K_{GridDepth} = K_{RgbGrid}^{-1} \times K_{RgbDepth}$$

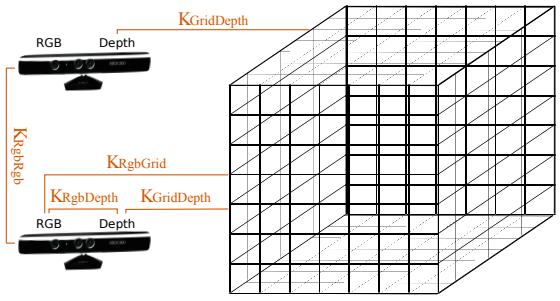


Fig. 1. Kinect grid calibration.

III. Classification of cells

A. HMM models

We want to distinguish mobile objects (chair, human beings) from static objects (walls). We define a cell containing a mobile object as being an 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 the classical occupancy grid method the state "empty" or "occupied" is calculated from the probability of occupation $P(r|C_i)$. We can consider the occupancy grid model as a two states HMM with no transitions between states as shown in Figure 2.

Fig. 2. Representation of the occupancy grid model (O: occupied; E: empty). Notice that $P(O) = 1 - P(E)$.

We modify the method of occupancy grid. For each cell, we use a three states HMM allowing to represent its dynamic and to determine its state. We remind that each cell is estimated independently of its nearby cells. The three states are:

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

In other words we can write:

- for mobile objects: $C_i = M$
- for statics objects: $C_i = O$
- for cells occupied by object: $C_i = M \vee O$

The representation of this HMM is shown in Figure 3.

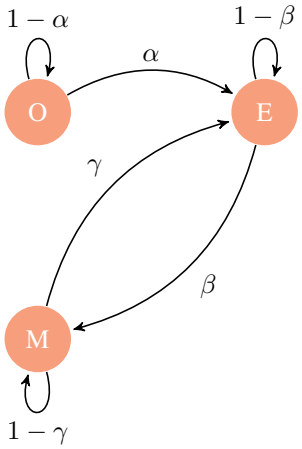


Fig. 3. The same HMM is used to model the evolution of each cell.

Probabilities of transition are $\alpha = 0.01$, $\beta = 0.1$ and $\gamma = 0.4$.

These probabilities respect the following assumptions :

- the detected cells in state O are regarded as being part of a wall. Consequently these cells have a weak probability of changing state. That's why the probability of transition of passing from O to another state is very weak. This probability is however not null because there is possibility that the model was mistaken since at the beginning it cannot make the difference between a wall and a mobile object which did not move yet (like a chair) ;
- when a cell appeared at least once free, there is no possibility for this cell to return to O because it is not a cell belonging to a wall. If not, this cell would have never passed in a free state ;
- the difference between a cell occupied in O and in M lies only in the passage or not in the state E that's why there is no transition between O and M.

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 $K_{GridDepth}$. 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 (Figure 5). The observation r (see the section II-A) 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.

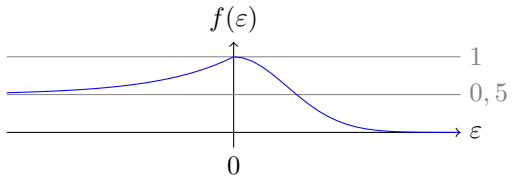


Fig. 4. Representation of occupation probability.

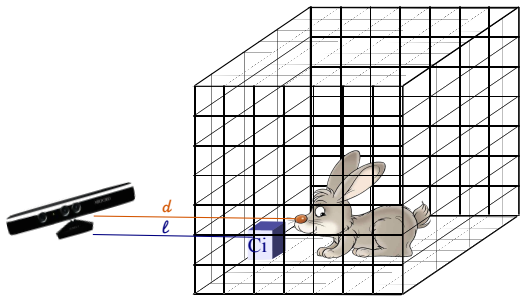


Fig. 5. d : distance between Kinect and object in occupancy grid, l : distance between Kinect and cells.

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 and ε_j the error of distance calculated with camera j.

C. Inference

To calculate the probability to be in one of the three states of HMM, we use to the Forward procedure [8]. We don't use the procedure Backward because we want the result to be given online.

The three observation functions are given by :

$$\begin{aligned} P(r|C_i = O) &= f(\varepsilon) \\ P(r|C_i = M) &= f(\varepsilon) \\ P(r|C_i = E) &= 1 - f(\varepsilon). \end{aligned}$$

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

We denote as po_t, pm_t, pe_t the probability of a cell being respectively occupied, mobile or empty at time t. We fix the initial probability with $po_0 = pe_0 = 0.5$ and $pm_0 = 0.0$.

IV. Identification of different mobile objects

In this part the aim is to gather the mobile cells belonging to the same object, so it will be possible to distinguish several persons in the same scene.

To gather the mobile cells we used the method «Component labelling» [9]. This method consists in assigning a label (a number) to each cell detected as mobile (state M of HMM). In Figure 6a), the colored cells are cells in state M. Thus in Figure 6b), the algorithm assign a different number to each colored cells. Then the technique is to look for each cell p if one of its neighbors has a smaller number. In this case, one assigns the number of the smallest neighbor to the cell p. This operation is shown in figure 6c). The cells at step t+2 take the smallest number among their neighbors at step t+1. One carries out this operation until there is no more change in the assignment of the cells labels. The Figure 6d) is the last step of the algorithm because the cells labels can't change. Thus all the cells having a same number will be gathered as being a same mobile object. In Figure 6d) the algorithm has detected two different objects, one object with cell carrying number 1 and another object with cell having number 7.

One key question in this algorithm is what is most judicious distance to consider for defining cell neighbors? Should we limit neighborhood at juxtaposed cells? Or is it necessary it to consider a wider range?

One of the problems if we consider only juxtaposed neighbors of the cell (neighborhood of size 1) is that it is possible that these neighbors aren't detected as mobile cell. As a consequence the leg or the arm can be cut of the body and the leg or the arm will be identified as an object separated from the rest of the body.

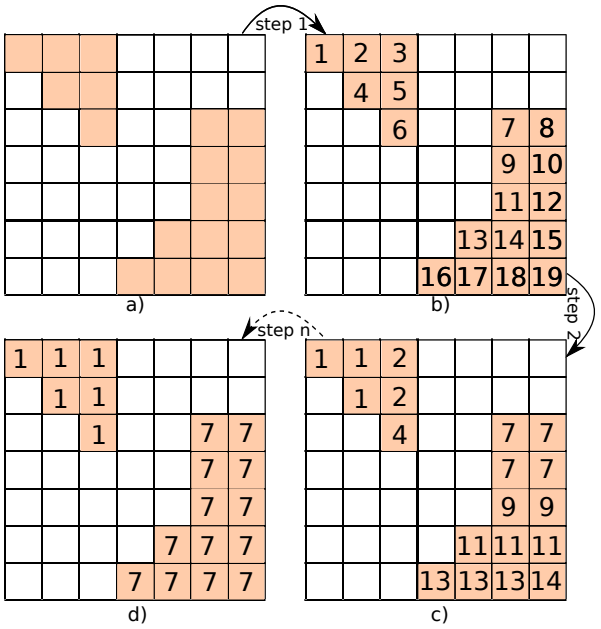


Fig. 6. a) mobile cells are colored, b) assignment of numbers to mobile cells, c) cells with number higher than it neighbor are modified, d) final result.

The second solution is to take farther cells in neighborhood, i.e. for cell C_i if we defined a neighborhood of size 2, the algorithm takes the neighbor cells positioned at C_{i-2}, C_{i-1}, C_{i+1} and C_{i+2}. The problem with this method is the risk of integrating cells which don't belong to the same object.

For illustrating these two methods we can give the following example. Figure 6 shows the case where we considered only juxtaposed neighbors cells. Two object in 6d) were detected. On the other hand if we had considered the neighbor of size 3 in Figure 6d), the two blocks of mobile cells would have gathered in only one object. In section V-B.2 we discuss this problem.

V. Results

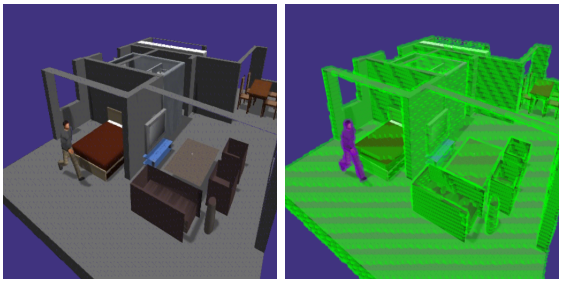
A. Simulation

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 7 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 propose to limit the quantitative evaluation to 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 spite of good specificity we can notice that there are as many false-positives (pixel detected as mobile whereas the pixel is static) as of true-positives (pixel detected as mobile and pixel is mobile). The problem comes from the fact that in the reference images only 2% of the pixels corresponded to the moving person whereas the other 98% were static objects or background. However visually the mobile points are always near to mobile object. The model has a some inertia and so if a pixel is mobile it takes a certain time before to return a static pixel. Table I shows the number of pixels for static and mobile objects obtained from the simulation and detected by our system.

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

TABLE I
NUMBER OF PIXELS IN EACH CATEGORY.



(a) Simulated apartment. (b) Distinction between static and mobile objects.

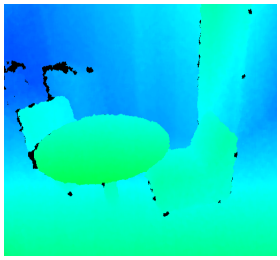
Fig. 7. Results in simulation.

B. Behavior in a smart room

We have tested our algorithm in an experimental apartment. Results presented here are qualitative. In Figure 8(a) and 8(b) we see the RGB and depth images. The image sent back by the Kinect is illustrated by Figure 8(c). Black spaces correspond to non reconstructed zones.

1) Results with one camera and one mobile object: We have tested the algorithm with one camera and one person walking in front of the camera. As illustrated by Figure 9(a), 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 (6 cm) and the uncertainty of the observation. 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) RGB image Kinect.

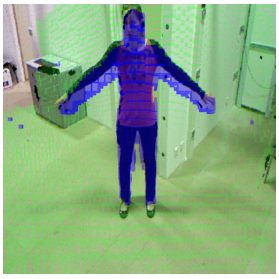


(b) Depth image Kinect.

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

Fig. 8. Image of Kinect camera.

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 9(a). 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.



(a)

(b)

Fig. 9. Green color: static objects. Blue color: mobile object.

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

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

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

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

Fig. 10. Chair becoming a mobile object.

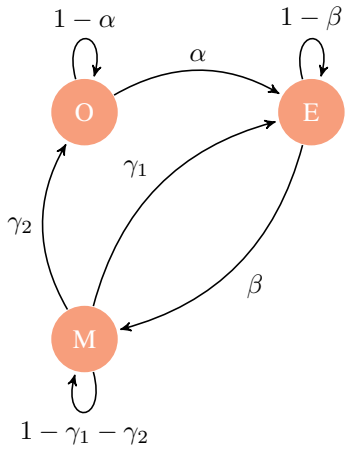


Fig. 11. Adding a link ($M \rightarrow O$) to the HMM so as to allow for a moved furniture to become a static object.

2) Results with one camera and several mobile objects: To test component labelling we realized a situation where several persons were in the field of view of the Kinect camera. The results of this test are showned in Figure 12(a). In this figure we haven't represented the occupancy grid, nor the scene but only the points representative of the center of mass of each cell of the object. Each group of points detected as one object is represented by a different color.

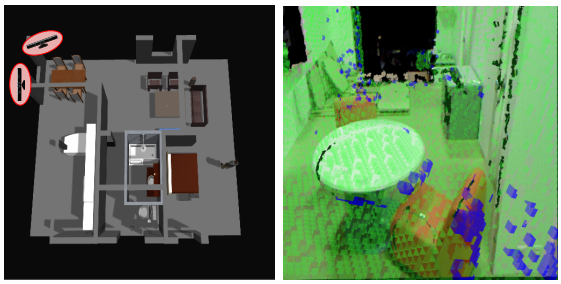
We said in section IV that to consider neighbor of size 1 is not a judicious choice because the parts of the body can be detected as separate objects. Thus to avoid this defect we increase the neighborhood to 3. But as before said the risk for taking a depth too large is to gather distinct object as shown in Figure 12(b). In this case two persons are in the room but they are too close and are considered as a single object (represented by a same color). When there is more space between the two persons as Figure 12(a), the algorithm correctly detected the two persons as separated mobile objects (as we can see because each object is in a different color).

3) Results with two cameras: To finish the experiment was then realized with two cameras placed as illustrated in Figure 13(a). We can see that the fusion of several cameras allows to discover more space while increasing the noise around static objects as illustrated in Figure 13. The noise is

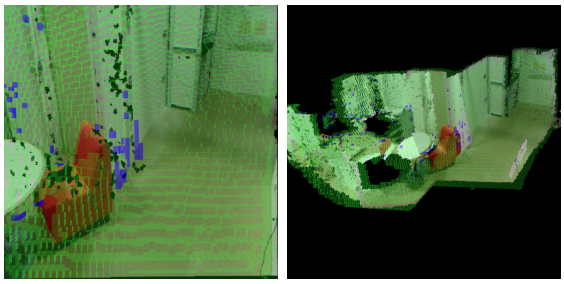
(a) Two persons are in the room and two objects are detected. (b) Two persons are in the room but only one object is detected.

Fig. 12. Results with component labelling.

due to interferences between the different infra-red images projected by the two camera Kinect.



(a) Position of the two cameras in the apartment. (b) View of one of the cameras.

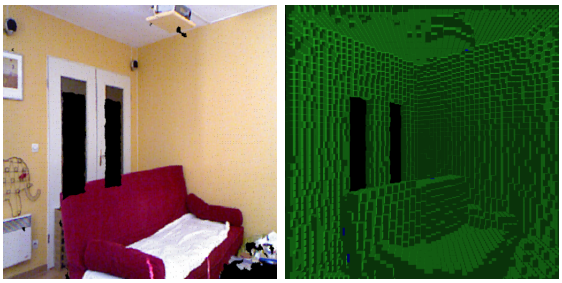


(c) View of the other cameras. (d) Fusion of the two cameras.

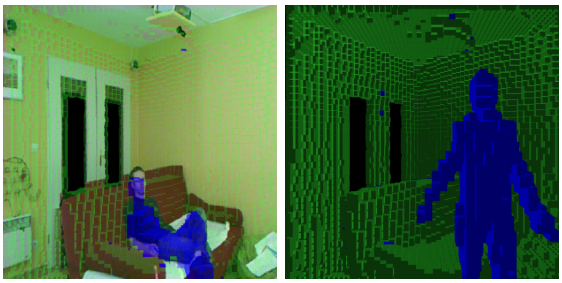
Fig. 13. Test with two cameras.

C. Behavior in realistic conditions

We have tested the algorithm, out of laboratory conditions, in a real 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(c) 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) which is the room where the test carryied out. We notice visually that the results are correct, the algorithm detects correctly the person.



(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).

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

VI. Conclusion

We presented a markerless system using Kinect cameras in the aim of tracking elderly people at home. First we proposed a system to merge several cameras by using a 3D 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. Thirdly, to solve the problem of distinguishing several persons in the field of view of the Kinect camera, we have implemented component labelling. This algorithm gather the cells belonging to the same object. 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. Concerning the detection of different persons the results are correct in most cases but the algorithm doesn't adress all the problems. For example, when two persons are too close the algorithm can't distinguish them. In the continuity of this work we will have to improve this part of the algorithm.

The aim of this project, as said in introduction, is to detect falls of elderly people. In this article, we presented the first

part of this project. It would be necessary in continuation of this work to learn characteristics of a person (as her size...) so as to be able to recognize her, track her and detect her activity (sitting, standing...). The purpose is to learn the habits of a person for thus detecting when an unusual behavior occurs (for example lying on the ground).

VII. Acknowledgement

This work has been partly funded by Region Lorraine. The authors thanks Abdallah Dib for his contribution to this work and Cedric Rose for his valuable comments.

References

[1] H. Bourdessol and S. Pin, *Prévention des chutes chez les personnes âgées à domicile*. France: éditions Inpes, 2005.

[2] a. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm." *Gait & posture*, vol. 26, no. 2, pp. 194–9, July 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17101272

[3] 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.

[4] A. Dubois, A. Dib, and F. Charpillet, "Using hmms for discriminating mobile from static objects in a 3D occupancy grid," in *Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI'11)*, 2011.

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

[6] 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.

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

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

[9] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," *Computer Vision and Image Understanding*, vol. 89, no. 1, pp. 1–23, Jan. 2003. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314202000309