Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Monocular 3D Head Tracking to Detect Falls of Elderly People

Caroline Rougier, Jean Meunier, Alain St-Arnaud and Jacqueline Rousseau

Abstract— **Faced with the growing population of seniors, Western societies need to think about new technologies to ensure the safety of elderly people at home. Computer vision provides a good solution for healthcare systems because it allows a specific analysis of people behavior. Moreover, a system based on video surveillance is particularly well adapted to detect falls. We present a new method to detect falls using a single camera. Our approach is based on the 3D trajectory of the head, which allows us to distinguish falls from normal activities using 3D velocities.**

I. INTRODUCTION

As other Western countries, Canada's population is growing older. According to the Public Health Agency of Canada [7], one Canadian out of eight was older than 65 years old in 2001. In 2026, this proportion will be one out of five. Moreover, in 1996, 93% of all seniors resided in private households, and among them, 29% lived alone [7]. Faced this reality, new technologies are developed to help them live in a more secure environment at home.

Falls are one of the most dangerous situation at home. Almost 62% of injury-related hospitalizations for seniors are the result of falls [8]. The first concerned are older people living alone because the situation can be aggravated if they cannot call for help, being unconscious or immobilized. Nowadays, the favored solution is to use wearable fall detectors like accelerometers or help buttons. However, older people often forget to wear them, and a help button is useless if the person is unconscious after the fall. Computer vision systems offer a new solution for fall detection which overcome these limitations.

Some research has been done to detect falls using image sensors. The easiest method to detect a fall is based on the shape of the person's silhouette or bounding box, but this method can be inaccurate, depending on the relative position of the person, camera, and perhaps occluding objects. Indeed, if the camera is placed sideways, the point of view can be affected by object occlusions. To overcome this problem, some researchers put the camera in the ceiling. For instance, Lee and Mihailidis [11] detect a fall using the shape of the person's silhouette, and Nait-Charif and McKenna [12] detect inactivity outside the normal zones of inactivity like chairs or sofas. Sixsmith and Johnson [15] use an infrared sensor to detect falls. Their system is based on 2D vertical velocity estimation which can be insufficient to discriminate a real fall from a person sitting down.

With respect to the approaches cited above, using 3D information reduces the effect of perspective foreshortening. In addition, this facilitates the use of wall-mounted cameras, which can cover large areas without having an overly wide field of view. Finally, we detect the fall while it is happening rather than after.

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

C. Rougier and J. Meunier are with the Department of Computer Science and Operations Research, Université de Montréal, Montreal, Canada (e-mail: rougierc@iro.umontreal.ca, meunier@iro.umontreal.ca).

A. St-Arnaud and J. Rousseau are respectively with the Centre de santé et de services sociaux Lucille-Teasdale and the Research Center of the Institut Universitaire de Gériatrie de Montréal, Montreal, Canada (e-mail: astarnaud@ssss.gouv.qc.ca, jacqueline.rousseau@umontreal.ca).

II. FALL DETECTION SYSTEM

The fall detection system is composed of a camera placed in a top corner of the room. The scene is analyzed and an alert is automatically reported when a dangerous situation is detected. To have a low-cost system, we use a USB webcam which gives rather low-quality images (compression problems, noise). Moreover, we use a wide angle of more than 70 degrees to see all the room, which unfortunately generates geometric distortions on images.

The main characteristics of our system are :

- *Tracking the head of the person :*
  We choose to track the head because it is usually visible in the scene (no occlusion problems), and has a large movement during a fall. We present the head representation and localization module in section III.
- *3D head trajectory extraction from a single camera :*
  2D tracking is not sufficient for a robust fall detection. So we decided to extract 3D information for higher efficiency. Usually, a multi-camera system is required to have precise 3D information, but we show that, with a single calibrated camera, we are able to track the head of the person in 3D. This method is described in section IV.
- *Fall detection based on velocity characteristics :*
  Having the head 3D trajectory, we can compute velocity characteristics which are useful for fall detection. The fall detection module is described in section V.

III. HEAD REPRESENTATION AND LOCALIZATION

Birchfield [1] showed that a head is well-approximated by an ellipse in the 2D image plane. So we choose to represent the head by a 3D ellipsoid which is projected as an ellipse in the 2D image plane (Fig. 1). To localize the head of the person, we use an iterative algorithm called *POSIT algorithm* [3] which takes as input arguments :

- The 3D dimensions of the head model which are defined with well-known anthropometric data [4].

- The 2D corresponding points projected in the image and corrected for distortion. The position of the ellipse in the image is determined by the tracking of an ellipse which is described in the next section.
- The intrinsic parameters of the camera, computed with the camera calibration toolbox for Matlab developed by J.Y. Bouguet [2]. This toolbox gives the focal length of the camera, the coordinates of the principal point and the image distortion coefficients (radial and tangential distortions).

The POSIT algorithm returns the relative position of the head in the camera coordinate system, that we can represented by a 4x4 homogenous matrix [6]. As we want to know this position in the world coordinate system attached to the XY ground plane, we must change the coordinate system using :

$$M_{Obj/World} = M^{-1}_{World/Cam}.M_{Obj/Cam}$$

$M_{World/Cam}$ is the known position of the world coordinate system in the camera coordinate system, $M_{Obj/Cam}$ the position of the head in the camera coordinate system computed by POSIT algorithm, and $M_{Obj/World}$ the desired position of the head in the camera coordinate system.

Thus, we can compute the pose of the head in a known coordinate system provided that we are able to track the head during the sequence of images. For this purpose, we use a particle filter to track an ellipse representing the head.

IV. 3D TRACKING

Several approaches have been considered to track the head along image sequences. Birchfield [1] proposed a simple tracker : the head is represented by an ellipse and updated by a local search using gradient and color information. Particle filters have been used also with success, for example to track a head with an ellipse [13] or a parametric spline curve [9] using color information or edge contours. Our approach is based on particle filters, also called "Condensation algorithm" (Conditional Density Propagation) [9], which allows abrupt variations on the trajectory and can deal with small occlusions.

The main idea of a particle filter is to approximate the probability density function of a system state by a weighted sample set (also called particles). The aim is to estimate the probability distribution $p(X_t|Z_t)$ given the state vector X_t of the tracked object and Z_t representing all the observations. This probability can be approximated by the set of N weighted samples $S_t = \{s^n_t, \pi^n_t, n = 1...N\}$. The three steps of the Condensation algorithm are :

1) *Selection :*
Select a new sample S_t by resampling from the old sample set S_{t-1} based on the sample weights π_{t-1}.
2) *Prediction :*
Predict new samples using a stochastic dynamical model.
3) *Measurement :*
Compute the new weights $\pi^n_t = p(z_t|X_t = s^n_t)$.

The mean state of the system is then estimated at time t using :

$$E[S_t] = \sum_{n=1}^{N} \pi^n_t.s^n_t$$

The existing methods using particle filters work well with small movements, but in our case, the movement can be very large (Fig. 1) because of the fall and the limited frame rate of the USB camera.

To overcome this problem using a particle filter, it would be necessary to have a lot of particles to find the new position of the head in the image. This would severely affect the computational performance and be incompatible with real time operation. Moreover, the position estimation may not be exact. So, we propose to adjust more judiciously the particles to track large movements like falls.

We suggest to use three particle filters with their own characteristics and purposes. Each particle of the filter is an ellipse, represented by a vector $X = [x, y, Hx, \rho, \phi]$ with (x, y) the center of the ellipse, Hx the length of the minor axis, $\rho = Hy/Hx$ the aspect ratio of the ellipse with Hy the length of the major axis, and ϕ the orientation of the ellipse. The weights of the particles are based on the Sobel gradient [5] around the ellipse perimeter. We also detect foreground locally [10] and verify that the ellipse delimits a foreground area. Finally, particles with a "reasonable" 3D pose are given higher weights.

The first particle filter is used to check if the head is just about the same place, so we just add a little noise on the position (x, y) of the ellipse. If the first filter does not succeed, a second filter is used to search farther away to find an new approximate position of the head using a more important noise. The third filter refines the previous position using weak noise on every parameters of the state vector. The principle of the algorithm is described in Fig. 2 and an example is shown in Fig. 3 showing its efficiency during a fall.

For each position of the ellipse, we compute the 3D pose of the head to obtain the 3D trajectory of the person's head. Then the 3D trajectory is analyzed to detect a fall.

V. FALL DETECTION

Wu [16] shows in a biomechanical study with wearable markers that falls can be distinguished from normal activities

Fig. 1. Head position (localized with an ellipse) during a fall on two successive images. In this case, the movement is about 70 pixels at 30Hz and image with size 640x480.

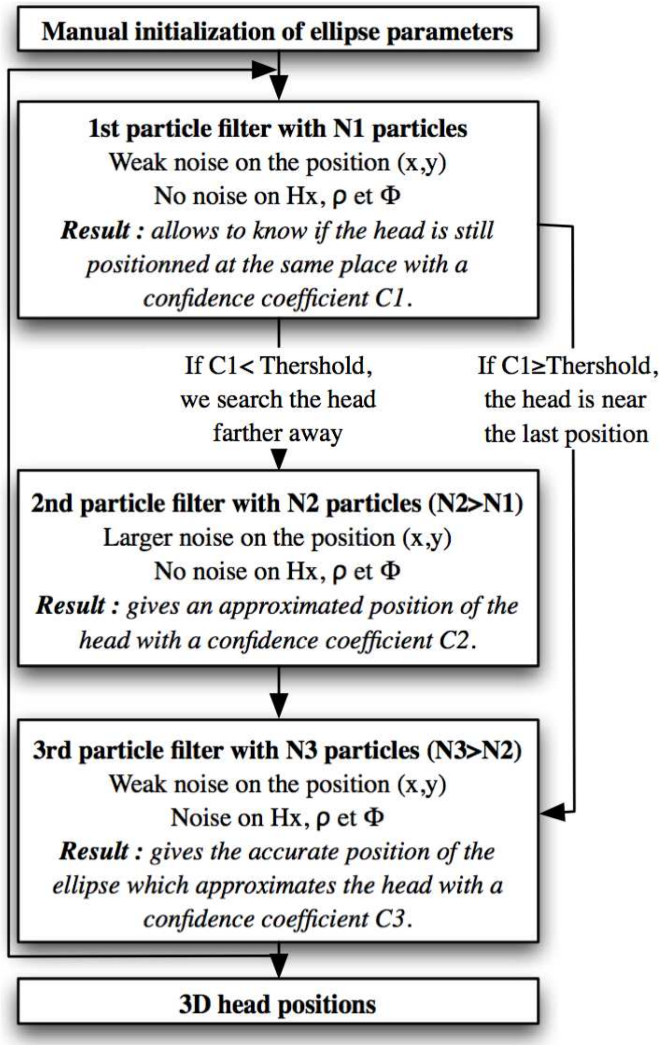

Fig. 2. Principle of the algorithm using 3 particle filters

Time $t-1$

1st filter

2nd filter

3rd filter

Time t

Fig. 3. Illustration showing the efficiency of the tracking during a fall. The green particles are the best particles of the filter, and the red all the others. The first filter looks around the last position at time $t-1$ of the ellipse . As it doesn't find the head, the second filter is used to search an approximated position of the head in the new image, and the third filter refines the position. The final ellipse is shown at time t.

using velocity characteristics. He proposes to use the vertical velocity V_v and the horizontal velocity V_h in the world coordinate system. Our 3D head trajectory extracted from the video is used to compute these characteristics with the aim of detecting a fall without requiring markers.

In Fig. 4, you can see the resulting 3D trajectory of a person with two interesting events : the first occurs when the person brutally sits down on a sofa, and the second is a fall which happens later when the person raises up again from the sofa.

We can differentiate a fall from normal activities using two appropriate thresholds for the vertical velocity V_v and for the horizontal velocity V_h. We also show an example of velocities obtained from the previous 3D trajectory in Fig. 4.

VI. EXPERIMENTAL RESULTS

Fall detection has been tested on 19 image sequences of daily normal activities and simulated falls. Nine sequences show different falls like forward falls, backward falls, falls when inappropriately sitting down, loss of balance. Ten sequences show normal activities like sitting down, standing up, crouching down. The image resolution is 640x480 pixels and the frame rate is 30 images per second.

The ellipse representing the head is manually initialized in the first image, and tracked through the sequences using the particle filters. The 3D head trajectory is computed for each sequence, and analyzed with the vertical velocity V_v and the horizontal velocity V_h. We recognize a fall activity if a peak occurs practically at the same time for both V_v and V_h, and if the negative peak V_v is less than $-1.5\ m/s$ and the positive peak V_h is more than $2\ m/s$. Table I shows the results obtained for our recognition system.

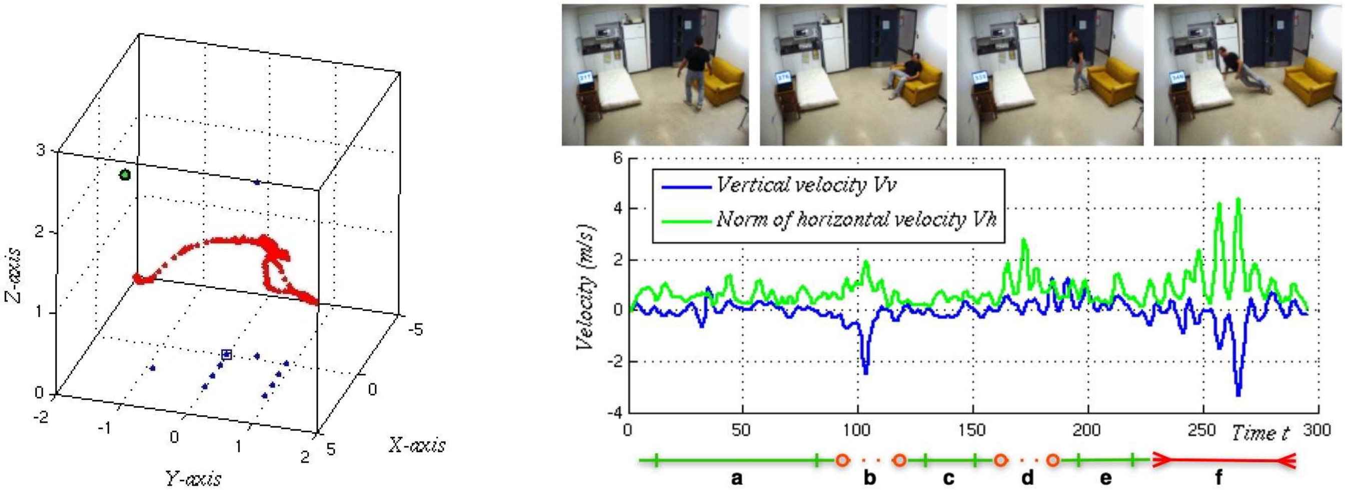

Fig. 4. On the left, the 3D trajectory obtained from image sequences. On the right, the vertical velocity V_v and the horizontal velocity V_h obtained without markers. We can see various actions including a fall. The different actions are : the person (a) stands up, (b) sits down, (c) is seated, (d) stands up again, (e) walks and (f) falls

	DETECTED	NOT DETECTED
FALLS	True Positive : 6	False Negative : 3
LURES	False Positive : 1	True Negative : 9

TABLE I
RECOGNITION RESULTS

Our recognition system gives encouraging results : a majority of normal activities are not detected as falls, and we detect 2 falls out of 3. The tracking sometimes loses the position of the head at the end of the fall, but this problem does not affect the fall detection system, because a fall is detected with the large velocities occurring at the beginning of the fall. Errors essentially occur when the person falls from an initial sitting position (head velocity is too low) or when the head tracking becomes unreliable.

VII. CONCLUSION AND FUTURE WORK

In this paper, a new method for fall detection is proposed. This study shows that we are able to extract the head 3D trajectory of a person using a single calibrated camera, and that this tracking can be useful for fall detection. The use of computer vision is indeed well-adapted to detect falls and allows to work with 3D velocity characteristics without any wearable sensor.

However, some points need to be improved. Currently, the ellipse representing the head in the first image is manually initialized. We will therefore develop an automatic method to detect the head when the person arrives in the room. We also plan to further improve the crucial part of the system, head tracking, to enhance our fall recognition system.

This feasibility study is currently working on Matlab ®, but a C++ version will be implemented soon using the OpenCV library [14]. In the future, our system will be tested with a larger number of video sequences with different view points and several persons.

REFERENCES

[1] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms", in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, June 1998, pp. 232-237.
[2] J.Y. Bouguet, "Camera calibration toolbox for Matlab" : *http://www.vision.caltech.edu/bouguetj/calibdoc/*.
[3] D.F. Dementhon and L.S. Davis, "Model-based object pose in 25 lines of code", *International Journal of Computer Vision*, vol. 15, no. 1-2, June 1995, pp. 123-141.
[4] L.G. Farkas, "Anthropometry of the head and face", Raven Press, New York, 2nd edition, 1994.
[5] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Prentice Hall, 2nd edition, 2002.
[6] R. I. Hartley and A. Zisserman, "Multiple view geometry in computer vision", Cambridge University Press, 2nd edition, 2004.
[7] Health Canada, "Canada's Aging Population", *Public Health Agency of Canada, Division of Aging and Seniors*, 2002.
[8] Health Canada, "Report on senior's falls in Canada", *Public Health Agency of Canada, Division of Aging and Seniors*, 2005.
[9] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", in *International Journal of Computer Vision*, vol. 29, no. 1, 1998, pp. 5-28.
[10] K. Kim, T.H. Chalidabhongse, D. Harwood and L. Davis, "Real-time foreground-background segmentation using codebook model", in *Real-Time Imaging*, vol. 11, no. 3, June 2005, pp. 172-185.
[11] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection", in *Journal of telemedicine and telecare*, vol. 11, no. 4, 2005, pp. 194-198.
[12] H. Nait-Charif and S.J. McKenna, "Activity summarisation and fall detection in a supportive home environment", in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 4, 2004, pp. 323-326.
[13] K. Nummiaro, E. Koller-Meier and L. Van Gool, "An adaptive color-based particle filter", in *Image and Vision Computing*, vol. 21, no. 1, January 2003, pp. 99-110.
[14] Intel Open Source Computer Vision Library : $http://www.intel.com/research/mrl/research/opencv/$
[15] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly", in *IEEE Pervasive Computing*, vol. 3, no. 2, April-June 2004, pp. 42-47.
[16] G. Wu, "Distinguishing fall activities from normal activities by velocity characteristics", in *Journal of Biomechanics*, vol. 33, no. 11, 2000, pp. 1497-1500.