3D Head Trajectory using a Single Camera

Caroline Rougier and Jean Meunier
Department of Computer Science and Operations Research (DIRO)
University of Montreal, QC, Canada
{rougierc,meunier}@iro.umontreal.ca

Abstract

Video surveillance applications need tools to track people trajectories. We present here a new method to extract the 3D head trajectory of a person in a room using only one calibrated camera. A 3D ellipsoid representing the head is used to track the person with a hierarchical particle filter. This method can run in quasi-real-time providing reasonable 3D errors for a monocular system.

Keywords: *computer vision, 3D, head tracking, monocular, particle filter, video surveillance.*

1. Introduction

1.1. Motivation

A 3D trajectory gives a lot of information for behavior recognition and can be very useful for video surveillance applications. For instance, in our research on fall detection [12], the 3D head trajectory of a person in a room is needed. Indeed, Wu [16] showed in a biomechanical study with wearable markers that the 3D head velocities were efficient to detect falls. In this work, an original 3D head tracker is proposed to recover the person location in a room with only one calibrated camera and no wearable markers. Our objectives are to extract the 3D head trajectory in real-time with sufficient precision to compute 3D velocities.

1.2. Related Works

Usually, a multi-cameras system is used to have some 3D information. Wu and Aghajan [17] have recovered both the 3D trajectory of the head and its pose with a multi-camera system without accurate calibration of the camera network. Their method was based on Chamfer matching refined with a probabilistic framework. Kawanaka et *al*. [6] have tracked the head in a 3D voxel space with four stereo cameras, using particle filtering with depth and color information. Kobayashi et et *al*. [8] have used AdaBoost-based cascaded classifiers to extract the 3D head trajectory using several cameras.

For a low-cost system, we prefer to use only one camera per room, which explains our interest in monocular methods. Very few attempts have been done to track the 3D head trajectory in real-time with a single camera. Hild [4] has recovered a 3D motion trajectory from a walking person, but he has supposed that the person was standing and that the camera optical axis was parallel to the (horizontal) ground plane. These assumptions can not be assumed in video surveillance applications: the camera need to be placed higher in the room

to avoid occluding objects and to have a larger field of view, and the person is not supposed to be standing or facing the camera.

1.3. Method Overview

Birchfield [1] showed that a head is well-approximated by an ellipse in the 2D image plane. Its tracker was based on a local search using gradient and color information. A head ellipse was also tracked by Nummiaro et *al.* [9] using a color-based particle filter. In our previous work [12], we also have used this idea to compute the 3D head localization from a 2D head ellipse tracked in the image plane. The 3D head pose was computed using a calibrated camera, a 3D model of the head and its projection in the image plane. This method worked well with a standing person, but the 3D pose can be wrongly estimated when the person falls (the ellipse ratio can change with the point of view).

In this work, we explore the inverse method: the head is seen as a 3D ellipsoid which is projected as an ellipse in the 2D image plane, and tracked through the video sequence using a particle filter. Section 2 describes the 3D head model and its projection in the image plane. The head tracking module using particle filter is shown in section 3 and is tested in section 4 with the HumaneEva-I dataset [14].

2. Head Model Projection

We chose to represent the head by a 3D ellipsoid, which is projected in the image plane as an ellipse [15]. The 3D head model will be tracked in the world coordinate system attached to the XY ground plane. To project the 3D head model in the image plane, we need to know the camera characteristics (intrinsic parameters) and the pose of the XY ground plane related to the camera (extrinsic parameters).

The intrinsic parameters can be computed using a chessboard calibration pattern and the camera calibration toolbox for Matlab [2]. The camera's intrinsic matrix K is defined by:

$$K = \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

with the focal lenght (f_x, f_y) and (u_0, v_0) in pixels. Notice that image distortion coefficients (radial and tangential distortions) are also computed and used to correct the images for distortion before processing with our methodology.

The extrinsic parameters are obtained from corresponding points between the real word and the projected image points. The plane-image homography obtained from these two sets of points is used to computed the extrinsic parameters:

$$M_{ext} = \begin{pmatrix} R & T \\ 0\ 0\ 0 & 1 \end{pmatrix} \quad (2)$$

Where R and T are respectively a 3D rotation matrix and a 3D translation vector, as described in the work of Zhang [18].

An ellipsoid is a quadric described by a matrix Q_C in the camera coordinate system such that $[x, y, z, 1]^T Q_C [x, y, z, 1]=0$ for a point (x, y, z) belonging to the ellipsoid. This ellipsoid is projected in the image plane with the projection matrix P as a conic C [11][15]:

$$C = Q_{C_{44}} Q_{C_{1:3,1:3}} - Q_{C_{1:3,4}} Q^T{}_{C_{1:3,4}} \tag{3}$$

The ellipse is described by the conic by $[u, v, 1]^T C [u, v, 1]=0$ for a point (u, v) in the image plane.

Concretely, the ellipsoid matrix representing the head in the head coordinate system has the form:

$$Q_H = \begin{pmatrix} \frac{1}{B^2} & 0 & 0 & 0 \\ 0 & \frac{1}{B^2} & 0 & 0 \\ 0 & 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{4}$$

With the semi-major A and the semi-minor B ellipsoid head axes.

The head ellipsoid is projected as Q_C in the camera coordinate system with the projection matrix P:

$$Q_C = P^{-1T} Q_H P^{-1} \tag{5}$$

The projection matrix P represents here the transformation from the head ellipsoid coordinate system to the image plane:

$$P = K M_{ext} M_{Head/World} \tag{6}$$

$M_{Head/World}$ will be defined during the tracking process by the translation and the rotation of the head in the world coordinate system (see eq. 9). Finally, the conic obtained from eq. 3 is used to define the ellipse parameters.

The different coordinate systems used for the ellipsoid projection are shown in Fig. 1.

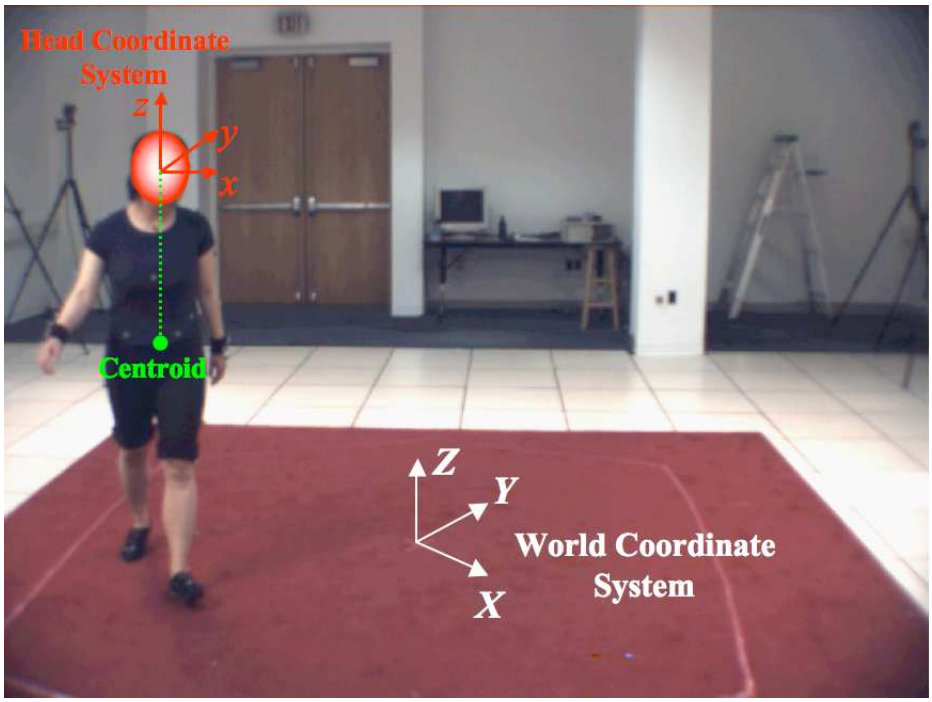

Figure 1. The different coordinate systems

3. 3D Head Tracking with Particle Filter

3.1. Particle Filter

Particle filters have been used with success in several applications, and in particular to track the head with an ellipse [9][12], or a parametric spline curve [5] using color information or edge contours. In our work, a particle filter is particularly suitable as it allows abrupt variations of the trajectory and can deal with small occlusions.

The main idea of particle filters is to estimate the probability distribution $p(X_t|Z_t)$ of the state vector X_t of the tracked object given Z_t, representing all the observations. This probability can be approximated from a set of N weighted samples (also called particles) defined by $S_t = \{s_t^{(n)}, \pi_t^{(n)}, 1 \le n \le N\}$. The main steps of the particle filter algorithm are:

- **Selection**
  N new ellipsoids are randomly selected from the previous elipsoids by favoring the best ones.
- **Prediction**
  The new ellipsoids are predicted with a stochastic dynamical model :

$$S_t'^{(n)} = A\, S_t^{(n)} + B\, w_t^{(n)} \tag{7}$$

Where $w_t^{(n)}$ is a vector of standard normal random variables, A and B are, respectively, the deterministic and stochastic components of the dynamical model.

- **Measurement**
  A weight $\pi_t^{(n)}$ is assigned to each ellipsoid (see Sect. 3.2}).
- **Estimation**
  The mean state is then estimated from the weighted ellipsoids.

$$E[S_t] = \sum_{n=1}^{N} \pi_t^{(n)} s_t^{(n)} \tag{8}$$

In our work, each particle of the filter is an ellipsoid, represented by the state vector:

$$X = [X_e, Y_e, Z_e, \theta_{Xe}, \theta_{Ye}] \tag{9}$$

Where (X_e, Y_e, Z_e) is the 3D head ellipsoid centroid expressed in the world coordinate system (translation component of the matrix $M_{Head/World}$, and (θ_{Xe}, θ_{Ye}) are respectively the rotation around the X and the Y axes (rotation component of the matrix $M_{Head/World}$). Figure 2 shows the parameters of our 3D ellipsoid model. Notice that two angles (instead of three) are sufficient to define the position and orientation of the ellipsoid since its minor axes have both the same length in our model.

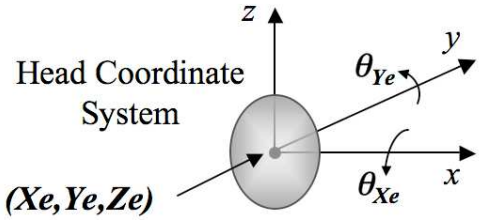

Figure 2. The 3D ellipsoid model

3.2. Particle Weights

The particle weights are based on foreground, color and body coefficients.

- **Foreground coefficient C_F**

  To compute the foreground coefficient, we first need to extract the person silhouette in the image. For this purpose, we use a background subtraction method, which consists in comparing the current image with an updated background image. We use the codebook method from the work of Kim et al. [7], which takes into account shadows, highlights and high image compression. An example of foreground silhouette is shown in Fig. 3.

(a) Original image (b) Foreground silhouette

Figure 3. From the current image (a), the foreground silhouette (b) is extracted with the codebook background subtraction method [7].

The foreground coefficient is computed by searching for silhouette contour points along N_e ellipse normals. For each normal, the distance d_e from the ellipse point to the detected silhouette point is used to compute the foreground coefficient:

$$C_F = \frac{1}{N_e}\sum_{n=1}^{N_e}\left(D_e(n) - d_e(n)\right)/D_e(n) \quad , \qquad C_F \in [0\ldots1] \tag{10}$$

Where D_e, the half-length of the normal, is used to normalize the distances.

Some examples of foreground coefficients, obtained for a projected ellipse, are shown in Fig. 4. From the projected ellipse shown in green, N_e ellipse normals (in blue) are used to detect the foreground silhouette points (in red).

$C_F = 0.89$ $C_F = 0.43$

Figure 4. Examples of foreground coefficients.

- **Color coefficient C_C**

The color coefficient is based on the normalized 3D color histogram of the head ellipse [9]. The color histogram, in the RGB color space, is composed of $N_b = 8x8x8$ bins and is computed inside a rectangular zone included in the ellipse. The comparison is done by computing the histogram intersection between an updated color head model and the target model:

$$C_C = \sum_{i=1}^{N_B}\min\left(H(i), H_{ref}(i)\right) \quad , \qquad C_C \in [0\ldots1] \tag{11}$$

- **Body coefficient C_B**

The body coefficient is used to link the head to the body through the body center. The projection of the 3D point corresponding to the centroid of the person should be near the centroid of the 2D silhouette (distance d_b compared to the half-major axis of the bounding box D_b). This coefficient is only used when the bounding box is valid (not used in case of occlusion for example).

$$C_B = (D_b - d_b)/\mathrm{D_b} \quad , \qquad C_B \in [0 \ldots 1] \tag{12}$$

- **Final coefficient**

The final ellipsoid coefficient is an amplified combination of these three coefficients giving larger weights to the best particles:

$$C_{final} = \frac{1}{\sqrt{2\pi}\sigma} \exp^{(\mathrm{C_F C_C C_B})/2\sigma^2} \qquad \text{with } \sigma = 0.25 \tag{13}$$

Each coefficient has an important role in the head tracking. The foreground coefficient is useful for the 3D localization precision when the ellipse well matched the foreground silhouette. The color coefficient is used to prevent the ellipsoid to hang on to something else inside the silhouette, and the body coefficient helps to prevent an abnormal rotation of the ellipsoid around its center.

3.3. Tracking

Initially, a head detection module is used to calibrate our system. When the person stands up, the top head point is detected in the foreground silhouette. From this point, several 2D ellipses are tested and the one, which has the biggest foreground coefficient C_F, is kept. When $C_F > 0.7$, the ellipse is supposed sufficiently reliable. This head ellipse is used to calibrate the ellipsoid proportion related to the human height (The height H of the person is supposed to be known and the aspect ratio of the ellipse is fixed at 1.2 for a human head [1]). Finally, to obtain the initial 3D position of the ellipsoid, we use a floor-image homographic transformation of the image position of the feet to get the (X, Y) location and the person height H to infer the Z coordinate.

To have a reliable 3D head localization, we need to precisely estimate the head projection in the image. With a conventional particle filter, a lot of particles are needed for precision, which severely affects the computational performance and is incompatible with real-time operation. We rather prefer using an improved particle filter based on a hierarchical scheme with several layers, similar to the annealed particle filter [3]. We chose to work with 250 particles and 4 layers, which is a good compromise between performance and computational time. The known velocity between two successive images is added to the particles to predict the next 3D ellipsoid localization before propagating the particles.

Each layer has a different stochastic component for the model propagation:

$$B = [B_{X_e}, B_{Y_e}, B_{Z_e}, B_{\theta_{Xe}}, B_{\theta_{Ye}}] \tag{14}$$

The stochastic component is sufficiently large for the first layer and decreases for the next layers, such as $B_{l+1}=B_l/2$ with l the current layer and $l+1$ the next layer. At the beginning, to refine the initial head position, B will be large for the X and Y components (initially, the person is supposed to be standing up, so that Z_e is approximately known and, θ_{Xe} and θ_{Ye} are close to zero). For the next images, B is reinitialized from the current velocity between two successive images and fixed to *0.1m* or *0.1 rad* if the velocity is too small.

Figure 5 shows some examples of 3D head tracking for a running person analyzed at 10Hz. The first layer '*Layer 1*' shows the large diffusion of the particles at the beginning directed towards the movement. Then, the diffusion decreases until the final layer, which allows to obtain a well matched 3D ellipsoid.

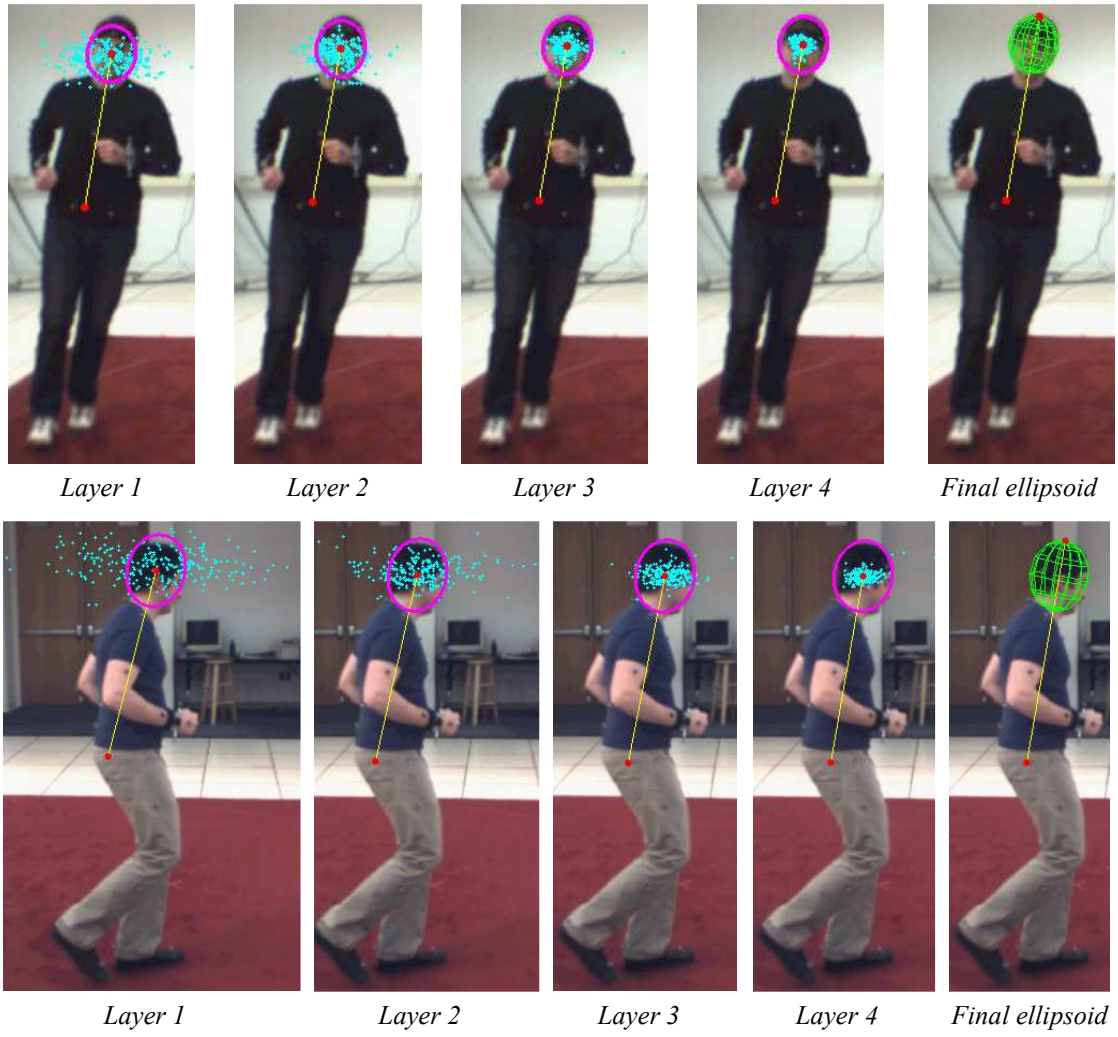

Figure 5. Examples of 3D head tracking. The images show the particles and the mean state ellipsoid obtained for each layer and the resulting ellipsoid.

Our hierarchical particle filters with 4 layers and 250 particles is a good compromise between tracking precision and reasonable computational time. In a more recent work, we have shown the ability of our 3D head tracker to deal with large motion occurring during a fall [13]. The resulting 3D head trajectory can then be used for fall detection.

4. Experimental Results

Our method is implemented in C++ using the OpenCV library [10] and can run in quasi-real time (*130ms/frame* on an Intel Core 2 Duo processor (2.4 GHz), not optimized code and image size of 640x480). The precision of our algorithm is evaluated with the HumanEva-I dataset [14], which contains synchronized multi-view video sequences and corresponding MoCap data (ground truth 3D localizations). As our method works with a single camera, we compare the results obtained from three different viewpoints (color cameras C1, C2 and C3) using the video sequences of three subjects (S1, S2 and S3). The HumanEva dataset contains several actions. We used the motion sequences "Walking" and "Jog" to evaluate our 3D trajectories at *30Hz*, *20Hz* and *10Hz*.

Figure 6 gives an example of 3D head trajectories obtained from different viewpoints. This figure shows that the curves are similar to the MoCap trajectory. The *X*, *Y* and *Z* curves obtained with these trajectories are shown in Fig. 7. The error is mainly a depth error in *X* and *Y* location. The Z location is rather well estimated and it is even possible to see the periodic motion of the walking person.

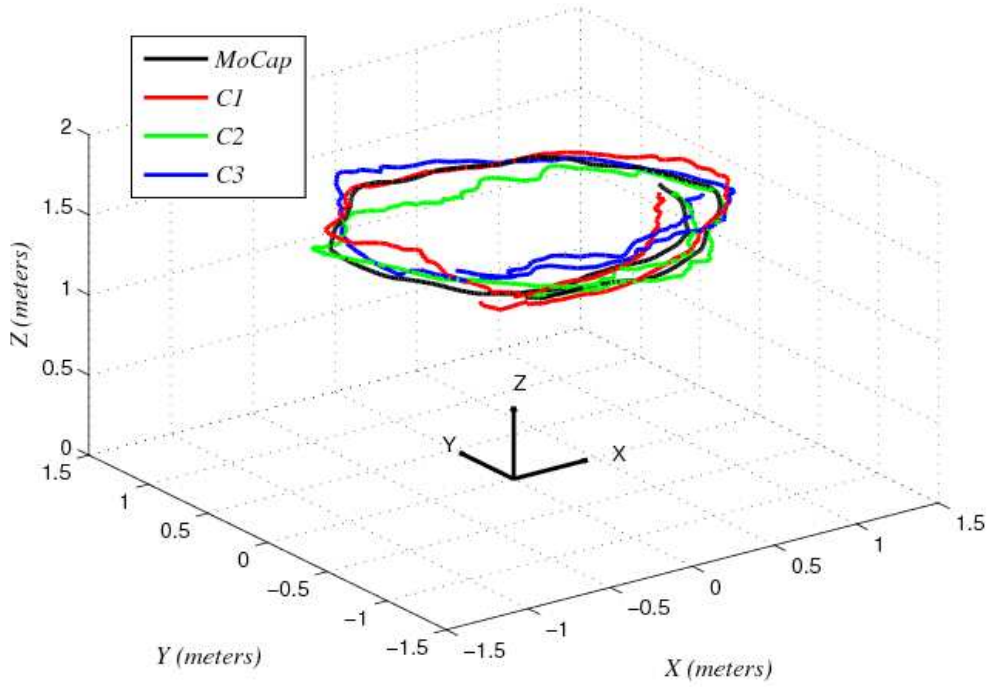

Figure 6. 3D head trajectories for a walking sequence of subject S1 (20Hz).

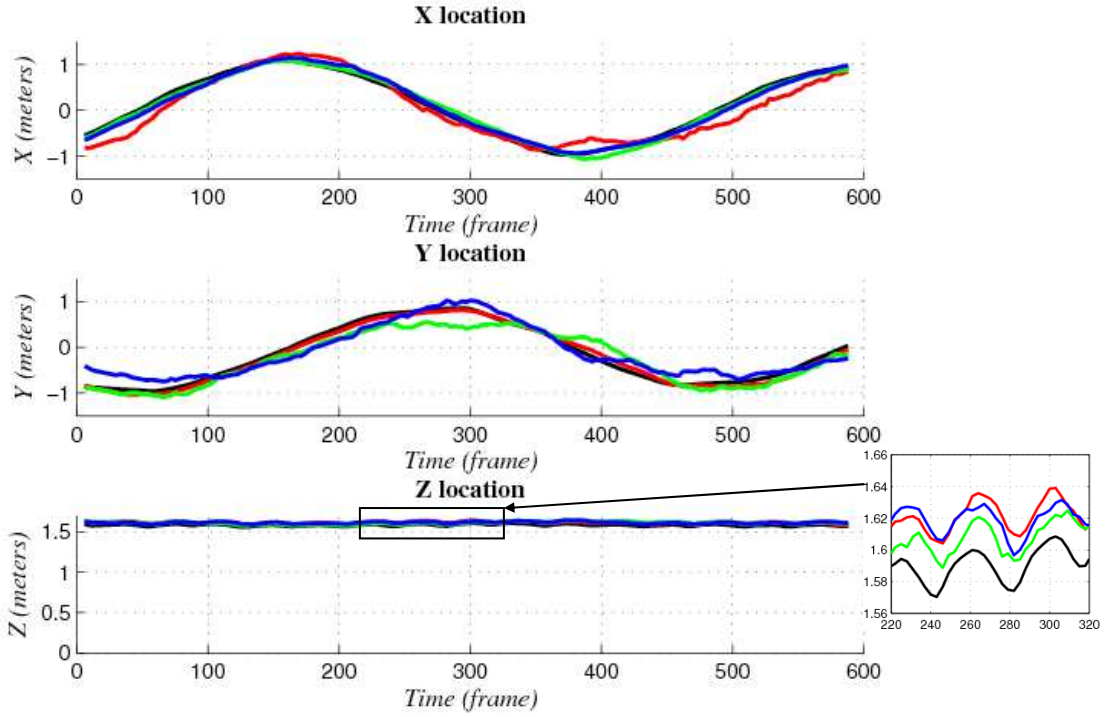

Figure 7. *X*, *Y* and *Z* curves obtained for the walking sequence of subject S1 (20Hz).

Table 1 summarizes the 3D mean errors obtained for each subject and each camera.

Frame rate	Camera	Action "Walking"			Action "Jog"		
		S1	S2	S3	S1	S2	S3
30Hz	C1	23.4±15.6	23.2±9.2	23.6±12.6	17.8±9	25.7±18.2	22.3±16
	C2	17.5±14.6	24±9.5	21.4±11	27±11.6	26±9	17±9
	C3	21±13.7	24.1±9.8	22.9±13.6	27±12.1	21.5±9.4	33.1±21.7
20Hz	C1	23.4±14.2	22.9±9.3	24.1±11.9	21.4±14.5	28.8±22.3	26.4±21.1
	C2	20.7±11.4	22.9±8.4	19.5±9	30±11.6	29.5±12.4	18.8±13.6
	C3	19.7±9.8	23.1±10.8	24.6±13.1	23.4±8.7	25.8±12.1	30.9±17.6
10Hz	C1	22.6±12.3	24.1±12.8	21.2±15.6	20.7±16.8	30.8±24.4	51.3±45.7
	C2	33.3±15.9	23.1±7.7	21.2±15.3	35.6±15.4	34.4±15.4	37.8±23.5
	C3	21.7±14	20.4±9.5	25.7±12.7	18.5±8.2	35.3±12.5	39.2±28.9

Table 1. Mean 3D errors (in cm) obtained from walking and jog sequences for different subjects (S1, S2, S3) and several view points (C1, C2, C3).

The 3D mean errors are about *5%* at a *4* to *6* meters distance. The similar mean errors for the different viewpoints demonstrate that our method is view independent. As expected, when the movement is larger, the error tends to be slightly higher, but the 3D ellipsoid remains well tracked. Notice that our 3D head tracker is automatically initialized with a good 2D detected head ellipse and can deal with body occlusions (e.g. chairs or entry in the scene). However, a well-defined and not occluded head is necessary.

5. Conclusion

In this paper, we have shown that a 3D head trajectory can be computed with only one single calibrated camera. The method gave similar results for different viewpoints, different frame rates and different subjects. The 3D locations were estimated with a reasonable mean error of around 25cm (5% at 5 meters), which is sufficient for most activity recognition based on trajectories.

Acknowledgement

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

References

[1] S. Birchfield, “Elliptical head tracking using intensity gradients and color histograms”, Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 1998, pp. 232-237.

[2] J. Bouguet, “Camera calibration toolbox for matlab”, 2010

URL: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

[3] J. Deutscher, A. Blake and I. Reid, “Articulated body motion capture by annealed particle filtering”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2., 2000, pp. 126-133.

[4] M. Hild, “Estimation of 3d motion trajectory and velocity from monocular image sequences in the context of human gait recognition”, International Conference on Pattern Recognition (ICPR), Vol. 4, 2004, pp. 231–235.

[5] M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking”, International Journal of Computer Vision, vol. 29, no. 1, 1998, pp. 5-28.

[6] H. Kawanaka, H. Fujiyoshi and Y. Iwahori, “Human head tracking in three dimensional voxel space”, International Conference on Pattern Recognition (ICPR), Vol. 3, 2006, pp. 826-829.

[7] K. Kim and T. Chalidabhongse, D. Harwood, L. Davis, “Real-time foreground-background segmentation using codebook model”, Real-Time Imaging, vol. 11, no. 3, 2005, pp. 172-185.

[8] Y. Kobayashi, D. Sugimura, K. Hirasawa, N. Suzuki, H. Kage, Y. Sato and A. Sugimoto, “3d head tracking using the particle filter with cascaded classifiers”, Proc. of British Machine Vision Conference, 2006, pp. 37-46.

[9] K. Nummiaro, E. Koller-Meier and L. Van Gool, “An adaptive color-based particle filter”, Image and Vision Computing, Vol. 21, No. 1, January 2003, pp. 99-110.

[10] OpenCV: Intel open source computer vision library, 2010

URL: http://opencv.willowgarage.com/wiki

[11] R.I. Hartley and A. Zisserman, “Multiple view geometry in computer vision”, 2nd edn. Cambridge University Press, 2004.

[12] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau, “Monocular 3d head tracking to detect falls of elderly people”, International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2006, pp. 6384-6387.

[13] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau, "3d head tracking using a single calibrated camera", Image and Vision Computing, 2010 (Submitted).

[14] L. Sigal, A. Balan and M.J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion", International Journal of Computer Vision (IJCV), Vol. 87, No. 1-2, March 2010, pp. 4-27.

[15] B. Stenger, P.R.S. Mendonça and R. Cipolla, "Model-based hand tracking using an unscented kalman filter", Proc. British Machine Vision Conference (BMVC), Vol. 1, September 2001, pp. 63-72.

[16] G. Wu, "Distinguishing fall activities from normal activities by velocity characteristics", Journal of Biomechanics, Vol. 33, No. 11, 2000, pp. 1497-1500.

[17] C. Wu and H. Aghajan, "Head pose and trajectory recovery in uncalibrated camera networks - region of interest tracking in smart home applications", ACM/IEEE International Conference on Distributed Smart Cameras, 2008, pp. 1-7.

[18] Z. Zhang, "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, 2000, pp. 1330-1334.

Authors

Caroline Rougier received her M.Sc. degree in image processing (2003) from the University of Paris 6, France, and her PhD degree in computer science (2010) from the University of Montreal, Canada. Her research interests include computer vision, image and video analysis and human activity recognition.

Jean Meunier received his BSc degree in physics from the Université de Montréal in 1981, his MScA degree in applied mathematics in 1983, and his PhD in biomedical engineering in 1989 from Ecole Polytechnique de Montréal. In 1989, after postdoctoral studies at the Montreal Heart Institute, he joined the department of computer science at the Université de Montréal, where he is currently full professor and chair. He is also a regular member of the Biomedical Engineering Institute at the same institution. His research interests are in computer vision and its applications to medical imaging and health care.