Multiple cameras fall data set

Edouard Auvinet[1] Caroline Rougier[1] Jean Meunier[1]
Alain St-Arnaud[2] Jacqueline Rousseau [3]

[1] University of Montreal, QC, Canada
{auvinet, rougierc, meunier}@iro.umontreal.ca

[2] Health and social service center Lucille-Teasdale, QC, Canada
alain.starnaud.lteas@ssss.gouv.qc.ca

[3] Research Center of the Geriatric Institute,
University of Montreal, QC, Canada
jacqueline.rousseau@umontreal.ca

Technical Report Number 1350

July 8, 2010

Abstract

Faced with the growing population of seniors, developed countries need to develop new healthcare systems to help elderly people staying at home in a secure environment. Falls are one of the major risk for seniors living alone at home, causing severe injuries. Computer vision provides a new and promising solution for fall detection. The number of works on fall detection using computer vision increases in the last few years, and currently there is no easy way to compare the different algorithms. We present here a unique video data set which will be very useful for the scientific community to test their fall detection algorithms. This reports provides an overview of our video data set acquired from a calibrated multi-camera system. This video data set contains simulated falls and normal daily activities acquired in realistic situations.

1 Introduction

1.1 Context

Due to the growing population of seniors in developed countries, healthcare systems need to be develop to ensure the safety of elderly people at home. Falls are one of the major risks for seniors living alone at home, often causing severe injuries. Usually, wearable fall detectors like accelerometers [1, 2], gyroscopes [3] or help buttons [4] are used to detect falls. But seniors often forget to wear them, and a help button is useless if the person is unconscious after the fall. Moreover, these sensors need a battery regularly replaced or recharged for adequate functioning. Therefore, a new and promising solution for fall detection is the use of computer vision, as no sensors need to be worn with this technology. Moreover, a video camera gives many information about the person, but also about the person's environment. Indeed, it is possible to know where the person is in a room and what are its/her actions.

1.2 The fall detection problem

One of the key fall detection problems is to recognize a fall among all the daily life activities, especially sitting down and crouching down activities which have similar characteristics to falls (for example, a large vertical velocity). Noury *et al.* [5, 6] have proposed to decomposed the fall event in four phases:

- The *pre-fall phase* corresponding to daily life motions.
- The *critical phase*, corresponding to the fall, is extremely short. This phase can be detected by the movement of the body toward the ground or by the impact shock with the floor.
- The *post fall phase* where the person is generally motionless on the ground. This phase can be detected by a lying position or by an absence of motion.
- The *recovery phase*, eventually, is the person is able to stand up alone or with the help of another person.

1.3 Related Works

1.3.1 Monocular systems

One of the commonly used fall detection methods is to analyze the bounding box representing the person [7, 8, 9] in the image. This simple method works only if the camera is placed sideways, and can fail because of occluding objects. For more realistic situation, other researchers [10, 11] placed the camera higher in the room to avoid occluding objects and to have a larger field of view. Lee and Mihailidis [10] detect falls by analyzing the person silhouette and the 2D image velocity, with special threshold for usual inactivity zones like the bed or the sofa. Nait-Charif and McKenna [11] track the person with an ellipse, and the resulting trajectory is used to detect abnormal inactivities outside usual inactivity zones. Rougier *et al.* [12] use the Motion History Image to detect a possible fall when large motion occurs. A fall is confirmed by analyzing the ratio and orientation of the ellipse representing the person. The large human shape deformation during a fall can also be used to detect a fall [13, 14].

The vertical velocity was also used to detect falls by considering the 2D vertical image velocity [15] or the 3D vertical velocity [16, 17]. For example, using a calibrated camera, Rougier *et al.* [17, 18] use the 3D head velocity obtained by tracking an elliptical model of the head with a particle filter.

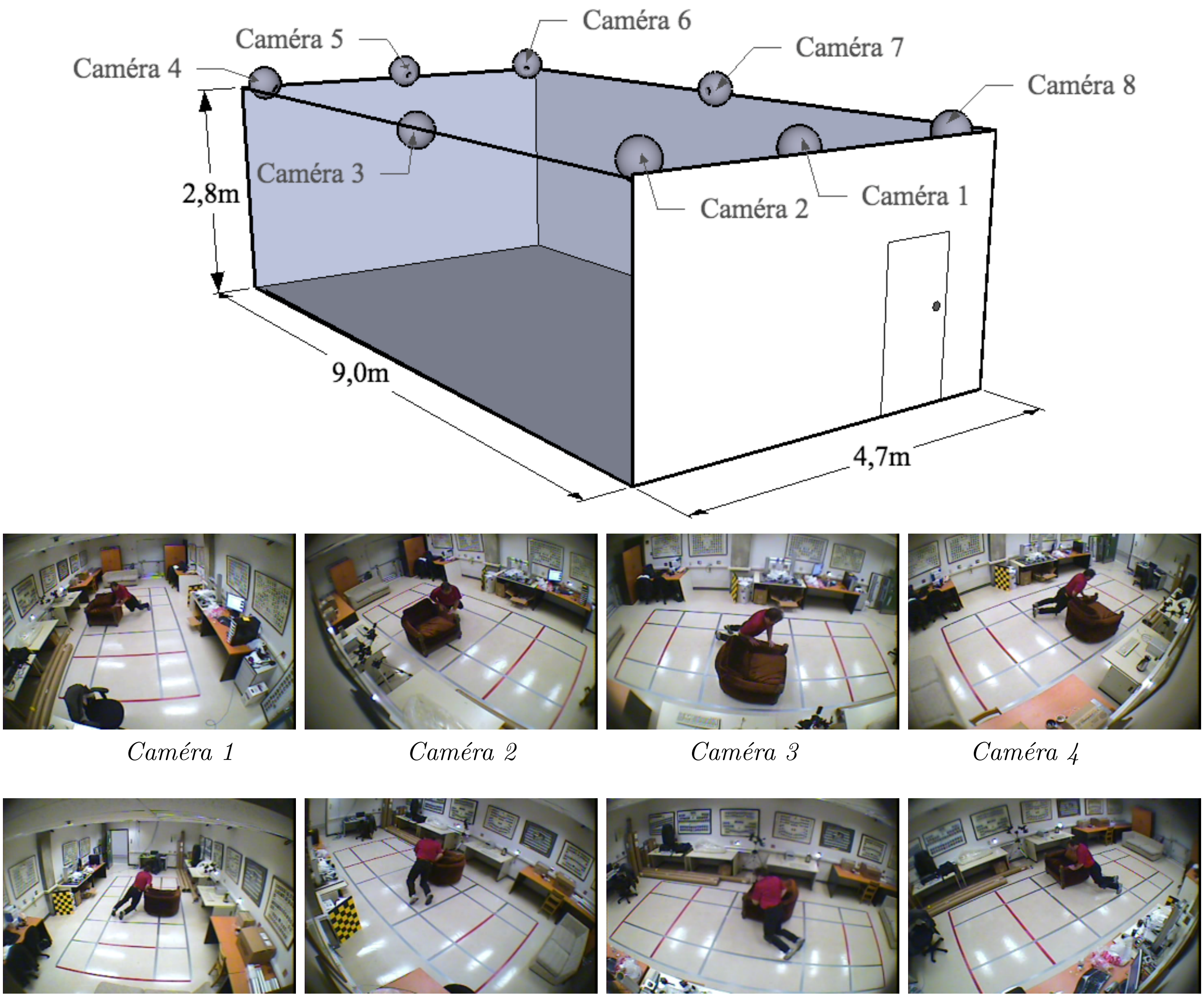
1.3.2 Multi-camera systems

With a calibrated multi-camera system, it becomes possible to extract a 3D shape of the subject which is very useful for fall detection. From the silhouettes extracted from each camera, Anderson *et al.* [19] reconstruct a three-dimensional representation of the human in voxel space. The fall detection step is performed by analyzing the states of the voxel person with a fuzzy hierarchy. A 3D shape of the person is also reconstructed by Auvinet *et al.* [20, 21] from a multiple cameras network. Their method consists of the combination of homographies (a homography is a transformation between projective planes) of the projected human silhouette (previously segmented with a foreground/background algorithm) on the ground and parallel planes for gathering information from different cameras and locate the person in the room. Fall events are detected by analyzing the volume distribution along the vertical axis, and an alarm is triggered when the major part of this distribution is abnormally near the floor during a predefined period of time, which implies that a person has fallen on the floor.

Thome and Miguet [22] use a Layered Hidden Markov Model (LHMM) to distinguish falls from walking activities. In their method, motion analysis characteristics are obtained from a metric image rectification in each view. With two uncalibrated cameras, Hazelhoff *et al.* [23] detect falls by analyzing the direction of the principal component and the variance ratio of the human silhouette obtained from a Principal Component Analysis (PCA). A head tracking module is used to improve their recognition results.

2 Data Description

Our multi-camera system is composed of eight inexpensive IP cameras (Gadspot gs-4600 [24]) with a wide angle to cover all the room as shown in Fig. 1.



Caméra 1 *Caméra 2* *Caméra 3* *Caméra 4*

Caméra 5 *Caméra 6* *Caméra 7* *Caméra 8*

Figure 1: Camera configuration and fall example viewed from all the cameras.

Our video sequences contained typical difficulties which can lead to segmentation errors like:

- **High video compression** (MPEG4) which can give artifacts in the image.
- **Shadows** and **reflections** which can be detected as moving objects during a segmentation process.
- **Cluttered and textured background**.
- **Variable illumination** which must be taken into account during the background updating process.
- **Carried objects** (bags, clothes, etc) which must also be taken into account during the background updating process.
- **Occlusions** (chairs, sofa, etc).
- **Entering/leaving** the field of view.
- **Different clothes** with different color and texture. Putting on and taking off a coat.

The video data set is composed of several simulated normal daily activities and falls viewed from all the cameras and perform by one subject. Some examples are shown in Fig. 2.

- **Normal daily activities** like walking in different directions, housekeeping, activities with characteristics similar to falls (sitting down/standing up, crouching down). But also image processing difficulties like occlusions or moving objects.
- **Simulated falls** like forward falls, backward falls, falls when inappropriately sitting down, loss of balance. Falls were done in different directions with respect to the camera point of view. Notice that a mattress was used to protect the person during the simulated falls.

3 Data analysis

The principal aim of this data set is the fall detection. We propose the following classification for fall event detected.

let t_{fall} be the time of fall, if an fall event is detected before this time, it is classified as false positive (FP), if it is detected after t_{fall}, it is then declared true positive (TP). If no fall event are detected after the t_{fall} time, a false negative (FN) is then counted. When no fall event are detected before t_{fall}, a true negative (TN) event can then be counted.

Statistical information representing results like sensitivity and specificity can the be computed.

$$sensitivity = \frac{TP}{TP+FN}$$
$$specificity = \frac{TN}{TN+FP}$$

forward fall *backward fall* *loss of balance*

loss of balance *loss of balance* *falls when sitting down*

a - Example of falls

carrying a box *occlusion* *putting off a coat*

sitting down *crounching down* *house keeping*

b - Example of normal activities

Figure 2: Examples of falls and normal daily activities.

4 Dataset Parameters

4.1 Camera calibration

Intrinsic parameters were computed using the chess board method[25] to define the focal distance $\mathbf{f} = (f_x, f_y)$, the optical center $\mathbf{c} = (c_x, c_y)$, the skew coefficient α and the radial distortion $\mathbf{k} = (k_1, k_2, k_3, k_4, k_5)$ as presented in [26]. The later parameters are necessary because of non negligible radial distortion due to the large field of view of the camera lenses. External parameters, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$ were calculated using feature points manually placed on the floor and the method from [27]. Altogether, those parameters define the projective camera model described as follows. Let $\mathbf{X} = (X, Y, Z)$ be the real world vector of a 3D point and $\mathbf{X}_c = (X_c, Y_c, Z_c)$ his coordinates in the camera c space then :

$$\mathbf{X}_c = \mathbf{R}\,\mathbf{X} + \mathbf{T}$$

The normalized image projection (x_n, y_n) is defined by:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$$

Where the tangential distortion vector (d_x, d_y) is :

$$\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} 2k_3\, x_n\, y_n + k_4\,(3x_n^2 + y_n^2) \\ k_3\,(x_n^2 + 3y_n^2) + 2k_4\, x_n\, y_n \end{bmatrix}$$

The normalized point coordinates (x_d, y_d) with radial distortion become :

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \left(1 + k_1 r_n^2 + k_2 r_n^4 + k_5 r_n^6\right) \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Finally, multiplying the normalized coordinates with the camera matrix gives pixel coordinates (x_p, y_p)

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_x & \alpha \cdot f_x & c_x \\ 0 & f_y & c_y \end{bmatrix} \begin{bmatrix} x_d \\ y_d \end{bmatrix}$$

camera's number	parameter	value
1	fx	471.326642112614593
	fy	426.662204436341881
	$u0$	346.657859672927259
	$v0$	218.359328644457833
	$alpha_c$	0.000000
	$kc1$	-0.397822369832092
	$kc2$	0.131226322783830
	$kc3$	-0.001152677509999
	$kc4$	-0.002291097372705
	$kc5$	0.000000000000000
	$r11$	0.700227670412089
	$r12$	0.693941023579491
	$r13$	0.167711255980950
	$r21$	0.517416584587631
	$r22$	-0.331432529694897
	$r23$	-0.788943950007675
	$r31$	-0.491895606387098
	$r32$	0.639216969466151
	$r33$	-0.591134822492788
	Tx	-3.476486050917116700
	Ty	-1.156083579992916384
	Tz	4.845474104553619327

camera's number	parameter	value
2	fx	472.958995562638279
	fy	428.846694149710117
	$u0$	345.469797776697703
	$v0$	248.497162677829976
	$alpha_c$	0.000000
	$kc1$	-0.389071900544632
	$kc2$	0.125991551945571
	$kc3$	0.001062638291482
	$kc4$	-0.000414066743725
	$kc5$	0.000000000000000
	$r11$	0.981072475018785
	$r12$	0.186485688334768
	$r13$	0.052152534040297
	$r21$	0.154336518165587
	$r22$	-0.590386623743320
	$r23$	-0.792227160393714
	$r31$	-0.116948868831810
	$r32$	0.785281301541856
	$r33$	-0.607993617999143
	Tx	-2.820809871433659737
	Ty	0.921080419583048752
	Tz	2.734747413825007243

camera's number	parameter	value
3	fx	467.134293415803540
	fy	422.904172514494576
	$u0$	332.870102520683645
	$v0$	273.174248857274392
	$alpha_c$	0.000000
	$kc1$	-0.416837423580768
	$kc2$	0.159970162292944
	$kc3$	0.000808858230169
	$kc4$	-0.001558735210223
	$kc5$	0.000000000000000
	$r11$	0.836278386413674
	$r12$	-0.542649442008455
	$r13$	-0.078549624475579
	$r21$	-0.427347841888337
	$r22$	-0.555314606236054
	$r23$	-0.713442015958044
	$r31$	0.343529158079374
	$r32$	0.630204150405890
	$r33$	-0.696297670799255
	Tx	-0.868917057630084059
	Ty	1.426262397558596604
	Tz	2.167801262810230583

camera's number	parameter	value
4	fx	467.401143838469864
	fy	422.921094825239720
	$u0$	370.510953156576420
	$v0$	235.676258506666187
	$alpha_c$	0.000000
	$kc1$	-0.407889633073927
	$kc2$	0.150978867978827
	$kc3$	-0.000569948689129
	$kc4$	0.000483970784893
	$kc5$	0.000000000000000
	$r11$	-0.046187
	$r12$	-0.99738
	$r13$	-0.055674
	$r21$	-0.55096
	$r22$	0.071925
	$r23$	-0.83143
	$r31$	0.83326
	$r32$	-0.0077272
	$r33$	-0.55283
	Tx	0.014083
	Ty	0.034027
	Tz	4.5147

camera's number	parameter	value
5	fx	468.525718027105029
	fy	424.278201738332086
	$u0$	356.867077527785284
	$v0$	260.554677252541751
	$alpha_c$	0.000000
	$kc1$	-0.413250949929596
	$kc2$	0.156961478481253
	$kc3$	-0.000470553455354
	$kc4$	0.000095303005574
	$kc5$	0.000000000000000
	$r11$	-0.08292
	$r12$	0.99656
	$r13$	-0.0015277
	$r21$	0.46107
	$r22$	0.037006
	$r23$	-0.88659
	$r31$	-0.88348
	$r32$	-0.074221
	$r33$	-0.46255
	Tx	-0.83742
	Ty	-0.25528
	Tz	5.8297

camera's number	parameter	value
6	fx	464.803932066310097
	fy	421.088099166200948
	$u0$	363.051656392719849
	$v0$	263.902505316842394
	$alpha_c$	0.000000
	$kc1$	-0.405432073000122
	$kc2$	0.141610268891392
	$kc3$	-0.000922297023790
	$kc4$	-0.000540244587381
	$kc5$	0.000000000000000
	$r11$	-0.724782926341326
	$r12$	-0.682829797858790
	$r13$	-0.091832330038109
	$r21$	-0.424800014868516
	$r22$	0.547827215705769
	$r23$	-0.788943950007675
	$r31$	0.542434011702633
	$r32$	-0.483351641031858
	$r33$	-0.687121920811728
	Tx	2.982910102426836602
	Ty	0.137181266696296547
	Tz	3.125268464727230366

camera's number	parameter	value
7	fx	471.672767285177940
	fy	426.985207175763549
	$u0$	373.002597127995728
	$v0$	216.597937176777009
	$alpha_c$	0.000000
	$kc1$	-0.397738076139422
	$kc2$	0.137287210994000
	$kc3$	-0.000105517334826
	$kc4$	0.001394449043772
	$kc5$	0.000000000000000
	$r11$	-0.997684327104262
	$r12$	-0.036044209156010
	$r13$	0.057678405290319
	$r21$	-0.067539297869570
	$r22$	0.625105171647650
	$r23$	-0.777612993475963
	$r31$	-0.008026624060089
	$r32$	-0.779707855139141
	$r33$	-0.626092033123340
	Tx	2.854633091998417967
	Ty	0.043378413955548417
	Tz	4.395608249803259241

camera's number	parameter	value
8	fx	470.561316433381705
	fy	425.594571513702249
	$u0$	355.172484892247837
	$v0$	231.057027802762576
	$alpha_c$	0.000000
	$kc1$	-0.412750560124808
	$kc2$	0.148158881927254
	$kc3$	-0.002749158333180
	$kc4$	-0.000425978486412
	$kc5$	0.000000000000000
	$r11$	-0.760617410767394
	$r12$	0.645702353338175
	$r13$	0.067302491269252
	$r21$	0.393016806454426
	$r22$	0.540500135718786
	$r23$	-0.743906844391378
	$r31$	-0.516719405753131
	$r32$	-0.539377487648044
	$r33$	-0.664885690578926
	Tx	1.736049071028928438
	Ty	-2.132541565128429738
	Tz	6.346925798481678612

4.2 Camera synchronisation

Due to difference of time needed for camera's stream to start, the event choosen to synchronise videos was the end event. This because, the recording programs where stopped all together. Delays are reported in annexed files for each scenarios.

Delays attached for each cameras equals to the number of frames of the cameras less the minimum of all camera's frames number.

For $Delay_{c,s}$ the delay of camera c in the scenario s and $Nframe_{c,s}$ we have then :

$$Delay_{c,s} = \min_{c\in\{1\cdots8\}} Nframe_{c,s}$$

Figure 3: Delay in frame for each camera versus scenario's number.

Scenario's number	camera 1	camera 2	camera 3	camera 4	camera 5	camera 6	camera 7	camera 8
1	3	3	8	4	23	6	6	0
2	25	40	0	16	18	33	33	6
3	12	16	8	16	35	20	20	0
4	72	79	78	0	68	82	83	56
5	17	24	5	11	18 7	26	28	0
6	0	100	106	90	89	103	104	89
7	28	14	16	0	1	17	18	20
8	92	79	0	81	64	81	82	56
9	18	9	1	19	13	11	12	0
10	14	15	19	33	12	17	19	0
11	23	4	20	14	0	6	7	12
12	21	6	13	8	0	3	7	0
13	16	33	0	7	27	27	36	13
14	49	36	38	0	29	29	7	14
15	15	19	19	15	34	40	23	0
16	23	29	0	2	12	9	3	3
17	21	26	15	0	10	0	29	18
18	99	105	86	0	84	108	109	77
19	19	27	16	19	5	29	0	20
20	25	9	3	10	10	4	5	0
21	20	30	22	3	8	33	32	0
22	0	46	51	41	53	46	47	34
23	31	52	52	45	54	60	50	0
24	3	36	7	0	37	10	33	1

4.3 Ground true

For all scenarios, the positions or movement of the subject have been tagged. The possible position noted are :

1	Walking, standing up
2	Falling
3	Lying on the ground
4	Crounching
5	Moving down
6	Moving up
7	Sitting
8	Lying on a sofa
9	Moving horizontaly

This is done for all frames and give the following position repertory

scenario's number	camera reference	period's start	period's end	position code
1	11	874	1011	1
		1012	1079	6
		1080	1108	2
		1109	1285	3

scenario's number	camera reference	period's start	period's end	position code
2	4	308	374	1
		375	399	2
		400	600	3

scenario's number	camera reference	period's start	period's end	position code
3	11	380	590	1
		591	625	2
		626	784	3

scenario's number	camera reference	period's start	period's end	position code
4	6	230	287	1
		288	314	2
		315	380	3
		381	600	6
		601	638	2
		639	780	3

scenario's number	camera reference	period's start	period's end	position code
5	11	288	310	1
		311	336	2
		337	450	3

scenario's number	camera reference	period's start	period's end	position code
6	1	325	582	1
		583	629	2
		630	750	3

scenario's number	camera reference	period's start	period's end	position code
7	6	330	475	1
		476	507	2
		508	680	3

scenario's number	camera reference	period's start	period's end	position code
8	4	144	270	1
		271	298	2
		299	380	3
scenario's number	camera reference	period's start	period's end	position code
9	11	310	472	1
		473	505	5
		506	576	7
		577	627	6
		628	651	2
		652	760	3
scenario's number	camera reference	period's start	period's end	position code
10	11	315	461	1
		462	511	5
		512	530	2
		531	680	3
scenario's number	camera reference	period's start	period's end	position code
11	7	378	463	1
		464	489	2
		490	600	3
scenario's number	camera reference	period's start	period's end	position code
12	11	355	604	1
		605	653	2
		654	770	3
scenario's number	camera reference	period's start	period's end	position code
13	4	301	430	1
		431	476	5
		477	525	7
		526	636	5
		637	717	8
		718	780	6
		781	822	6
		823	863	2
		864	960	3
scenario's number	camera reference	period's start	period's end	position code
14	6	372	555	1
		556	590	5
		591	856	8
		857	934	6
		935	988	6
		989	1023	2
		1024	1115	3

scenario's number	camera reference	period's start	period's end	position code
15	11	363	486	1
		487	530	5
		531	630	7
		631	754	6
		755	787	2
		788	870	3
scenario's number	camera reference	period's start	period's end	position code
16	4	380	455	1
		456	488	5
		489	530	4
		531	568	6
		569	629	5
		630	645	4
		646	670	6
		671	731	5
		732	817	7
		818	890	6
		891	940	2
		941	1000	3
scenario's number	camera reference	period's start	period's end	position code
17	6	251	315	1
		316	340	5
		341	361	4
		362	388	6
		389	410	5
		411	430	4
		431	460	6
		461	531	5
		532	620	7
		621	729	6
		730	770	2
		771	860	3
scenario's number	camera reference	period's start	period's end	position code
18	6	301	378	1
		379	430	5
		431	530	7
		531	570	6
		571	601	2
		602	740	3

scenario's number	camera reference	period's start	period's end	position code
19	10	255	498	1
		499	600	2
		601	770	3

scenario's number	camera reference	period's start	period's end	position code
20	11	301	544	1
		545	672	2
		673	800	3

scenario's number	camera reference	period's start	period's end	position code
21	11	408	537	1
		538	608	5
		609	794	7
		795	863	6
		864	901	2
		902	1040	3

scenario's number	camera reference	period's start	period's end	position code
22	1	317	586	1
		587	685	5
		686	737	7
		738	766	6
		767	808	2
		809	930	3

scenario's number	camera reference	period's start	period's end	position code
23	11	393	662	1
		663	688	5
		689	710	4
		711	744	6
		745	1519	1
		1520	1595	2
		1596	1661	6
		1662	1730	1
		1731	1769	5
		1770	1839	4
		1840	1886	6
		1887	2645	1
		2646	2698	5
		2699	2958	8
		2959	3035	6
		3036	3156	1
		3157	3237	5
		3238	3416	8
		3417	3573	6
		3574	3614	2
		3615	3745	6
		3746	3795	5
		3796	4042	4
		4043	4105	6
		4106	4204	1
		4205	4264	5
		4265	4440	7
		4441	4527	6
		4528	5200	1

scenario's number	camera reference	period's start	period's end	position code
24	6	350	974	1
		975	1315	1
		1316	1351	5
		1352	1414	4
		1415	1450	6
		1451	1750	1
		1751	1805	5
		1806	1844	4
		1845	1884	6
		1885	2490	1
		2491	2514	5
		2515	2563	4
		2564	2587	6
		2588	3040	1
		3041	3077	5
		3078	3125	6
		3126	3243	1
		3244	3353	1
		3354	3401	5
		3402	3500	4

References

[1] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[2] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, no. 2, pp. 285–291, 2008.

[3] A. Bourke and G. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering & Physics*, vol. 30, pp. 84–90, Janvier 2008.

[4] DirectAlert, "Wireless emergency response system." `http://www.directalert.ca/emergency/help-button.php`, 2010.

[5] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection - principles and methods," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 1663–1666, 2007.

[6] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, pp. 340–349, December 2008.

[7] B. Töreyin, Y. Dedeoglu, and A. Çetin, "HMM based falling person detection using both audio and video," in *Proceedings of the International Workshop on Computer Vision in Human-Computer Interaction (ICCV-HCI)*, vol. 3766, Octobre 2005.

[8] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2006)*, pp. 6388–6391, 2006.

[9] J. Tao, M. Turjo, M.-F. Wong, M. Wang, and Y.-P. Tan, "Fall incidents detection for intelligent video surveillance," *Fifth International Conference on Information, Communications and Signal Processing*, pp. 1590–1594, 2005.

[10] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of telemedicine and telecare*, vol. 11, no. 4, pp. 194–198, 2005.

[11] H. Nait-Charif and S. McKenna, "Activity summarisation and fall detection in a supportive home environment," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 4, pp. 323–326, 2004.

[12] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," in *IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, vol. 2, pp. 875–880, 2007.

[13] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Procrustes shape analysis for fall detection," in *ECCV 8th International Workshop on Visual Surveillance (VS 2008)*, 2008.

[14] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Occlusion robust video surveillance for fall detection," in *IEEE Transactions on Circuits and Systems for Video Technology*, (Submitted).

[15] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 42–47, 2004.

[16] G. Wu, "Distinguishing fall activities from normal activities by velocity characteristics," *Journal of Biomechanics*, vol. 33, no. 11, pp. 1497–1500, 2000.

[17] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," in *International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6384–6387, 2006.

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

[19] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Computer Vision and Image Understanding*, vol. 113, pp. 80–89, January 2009.

[20] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau, and J. Meunier, "Fall detection using multiple cameras," in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2554–2557, 2008.

[21] E. Auvinet, F. Multon, A. St-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3d silhouette vertical distribution," in *IEEE Transactions on Information Technology in Biomedicine*, (Submitted).

[22] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[23] L. Hazelhoff, J. Han, and P. H. N. de With, "Video-based fall detection in the home using principal component analysis," in *Advanced Concepts for Intelligent Vision Systems*, vol. 1, pp. 298–309, 2008.

[24] Gadspot, "Ip camera." `http://gadspot.com`, 2010.

[25] J. Bouguet, "Camera calibration toolbox for matlab," 2007.

[26] Heikkila and Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of the Conference Vision Pattern Recognition*, 1997.

[27] Z. Zhang, "A exible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 22(11):13301334*, 2008.