Coupling Fall Detection and Tracking in Omnidirectional Cameras

Barış Evrim Demiröz, Albert Ali Salah, and Lale Akarun

Department of Computer Engineering, Boğaziçi University
Istanbul, Turkey
{baris.demiroz,salah,akarun}@boun.edu.tr

Abstract. Omnidirectional cameras have many advantages for action and activity detection in indoor scenarios, but computer vision approaches that are developed for conventional cameras require extension and modification to work with such cameras. In this paper we use multiple omnidirectional cameras to observe the inhabitants of a room, and use Hierarchical Hidden Markov Models for detecting falls. To track the people in the room, we extend a generative approach that uses probabilistic occupancy maps to omnidirectional cameras. To speed up computation, we also propose a novel method to approximate the integral image over non-rectangular shapes. The resulting system is tested successfully on a database with severe noise and frame loss conditions.

1 Introduction

Most of the current state-of-the-art action recognition methods work on videos or images acquired from conventional perspective cameras. Increased popularity of cheap RGB-D sensors such as Kinect also drew the attention of researchers to this modality for indoor scenarios and short-range action recognition. Omnidirectional cameras, compared to conventional cameras and RGB-D cameras, cover more ground and may eliminate need for multiple cameras. This property of omnidirectional cameras makes them appealing to use for action recognition purposes, especially for indoor scenes. These camera systems were indeed designed primarily for monitoring purposes, albeit by human monitors [1]. For automatic action and activity recognition, methods developed for conventional cameras need to be extended or appropriately modified to function under the assumptions and conditions introduced by the use of omnidirectional cameras.

In this work, we extend the probabilistic person localization approach proposed in [2] to omnidirectional videos. To illustrate the usefulness of the approach, we test it on a fall detection application. The person's location and fall state (i.e. standing or fallen) is tracked using a Hierarchical Hidden Markov model (HMM), where the images are the observations. We show that the proposed method is robust to occlusions, noise and missing data, while incorporating the information coming from different cameras.

This paper is structured as follows. Section 2 summarizes publicly accessible action and activity recognition databases using omnidirectional cameras.

H.S. Park et al. (Eds.): HBU 2014, LNCS 8749, pp. 73–85, 2014.

Section 3 briefly describes related work in camera-based fall detection. Section 4 describes the tracking framework we extend in this paper, and Section 5 proposes a fast integration scheme over foreground silhouettes to speed up computations. Our experimental results are reported in Section 6, followed by our conclusions.

2 Public Action and Activity Recognition Datasets Using Omnidirectional Cameras

There are not many public action recognition datasets involving usage of omnidirectional cameras. The *Opportunity* dataset has 72 sensors in total from 10 modalities, and two of these sensors are omnidirectional cameras [3]. There are 12 subjects performing various morning activities like preparing coffee, eating a sandwich, etc.

The high-level activities (e.g. preparing a sandwich) and low-level actions (e.g. reaching) are annotated in parallel tracks. There are four different tracks of annotations, first track determines the high level activities like preparing a sandwich, the second track contains walk, stand, lie and sit actions. The remaining tracks define actions of right and left hands: reach, move, release, stir, sip, bite, cut, spread, respectively.

In a more recent study, Behera et al. introduced a new dataset of egocentric actions [4]. A fisheye camera was mounted on the chest of a person and two sets of data were collected. One set contains 27 videos of tool usage, and the other is a 23 video set of "labeling and packaging bottles" scenario. Along with the omnidirectional videos, there are head mounted depth and RGB recordings of the same scene. At the time of this writing, ground truth was not available, but to be released soon. In the first set (27 videos) nine actions are recorded: take/put box, pick hammer, take/put baton, take nail, hammer nail, put down hammer, pick screwdriver, driving screw and put down screwdriver. The second set is a "labeling and packaging bottles" scenario, 9 actions in 23 videos are: pick and put bottle, stick label, pick and put box, remove cover, put bottle inside box, take and put cover, write address, take and put tape dispenser, seal the box.

The BOMNI-DB dataset by Demiröz et al. includes two sets of videos recorded in a room for action recognition purposes with two omnidirectional cameras at the same time [20]. We use this publicly available database in our study. The first set is a single person scenario, where the person performs the following actions throughout the video: sitting, walking, drinking, washing hands, fainting, opening/closing door. The second set involves multiple people performing the actions: sitting, walking, standing, handshaking, being interested in object.

3 Related Work in Fall Detection

Prevention and early intervention of falls in elderly are critical in ambient assisted living, and there have been numerous methods developed for this purpose [5]. Wearable sensors are widely used for fall detection; there are studies

using sensors ranging from belts with pressure sensors to phones with 3-axis accelerometer [6,7]. On the other hand, cameras are less obtrusive and provide continuous observation. A broad survey of fall detection that includes vision modality among others is given in [8]. Here we focus only on visual approaches.

Most of the studies on camera-based fall detection in the literature depend on foreground segmentation, followed by an analysis of foreground height changes. In [9], motion history images from a single camera were used to quantify the motion of a person. But most approaches use more than a single camera, as the field of view imposes a limitation, and occlusions are always a consideration.

In an early approach, Nait-Charif and McKenna used omnidirectional cameras and modeled the person as an ellipse [10]. They have used particle filters to track persons, and considered unusual activities as cases of falling. In [11], wavelet transformation of the foreground's width-height ratio is used to model walking and falling using Hidden Markov Models (HMM). Audio features are used along with visual features to increase performance. Miaou et al. detected falls using width-height ratio of the single foreground area obtained [12].

Cucchiara et al. proposed a multi-camera approach for tracking a person and recognizing behaviors, including fall detection [13]. Using multiple cameras requires sophisticated tracking, particularly when the field of view of the cameras overlap. Omnidirectional cameras can sidestep this necessity.

In [14], the authors use two fixed, uncalibrated, perpendicular cameras. The main axis of the tracked object (i.e. the human) is computed, and used for inferring a fall. Similarly in [15], two cameras are used, independently, to detect the pose of a tracked person, and the motion information is integrated to detect falls. A recent approach by Kwolek and Kepski uses two Kinect cameras, one at the ceiling corner of the room, and the other at a one meter height [16]. This approach, like the conventional camera based approaches, is sensitive to camera placement to detect the fall events.

In the approach proposed in [17] the person is tracked and statistical outliers with respect to appearance and action are detected at different levels. Thus, an abnormal, horizontal position could be detected as a fall, if the spatial context is unexpected (e.g. on the floor, instead of the couch).

In a study using omnidirectional cameras, Huang et al. developed a rule based system that detects falls using the movement of the foreground segmentation over time [18]. They distinguish between radial falls (either inward or outward), and non-radial falls, as the appearance of the fall in an omnidirectional camera can vary depending on the action location. Over a dataset with 33 fall incidents, the authors report a 70 per cent detection rate. By injecting personal information like height, this rate can be increased to 80 per cent.

Contrasted with the related approaches, the method proposed in this paper has broad coverage in an indoor scenario, and is extremely robust to noise, occlusions and missing data. The next section describes the tracking methodology that forms the most important aspect of the approach.

4 Methodology

In this work, we track the inhabitants of an indoor scene using omnidirectional cameras and detect falls. We use multiple omnidirectional cameras to deal with occlusions. In [2], Fleuret et al. represented the ground plane with discrete, evenly spaced locations, and used a hidden location to handle transitions in and out of the scene. They have formulated a Bayesian framework to track multiple people, which utilizes the distance between the foreground segmentation image and a synthetic model of the human shape at any given location. Their model for a single person is an hidden Markov model (HMM), where the person's location is the hidden variable and images are observations. Our approach is built on Fleuret et al.'s foundations, and we extend their generative human shape model to fall detection in omnidirectional cameras.

Let L_t, F_t and I_t^c be a person's location, fall state of the person ($F_t = 1$ if fallen) and the image obtained from camera c at time t, respectively. Let bold letters indicate grouping of omitted indices, e.g. $\mathbf{I} = I_{1:T}^{1:C}$. At a given time, a person could be at one of the G locations, or at the hidden location that handles transitions in and out of the room. We are looking for the most likely value of the trajectory and the sequence of fall states of the person, given the observed images:

$$\{l_{1:T}^*, f_{1:T}^*\} = \underset{l_{1:T}, f_{1:T}}{\arg\max}\ \mathrm{P}(\mathbf{L} = l_{1:T}, \mathbf{F} = f_{1:T} \mid \mathbf{I}) \tag{1}$$

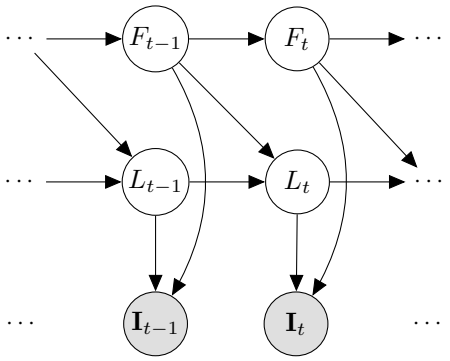

Fig. 1. The graphical model of the proposed system to detect falls. The variables F_t, L_t, I_t denote whether a person has fallen, location of the person and images obtained from all cameras at time t, respectively.

We can model the system using a Hierarchical HMM, given in Figure 1:

$$\mathrm{P}(F_t \mid F_{1:t-1}) = \mathrm{P}(F_t \mid F_{t-1}) \tag{2}$$

$$\mathrm{P}(L_t \mid L_{1:t-1}, F_{1:t-1}) = \mathrm{P}(L_t \mid L_{t-1}, F_{t-1}) \tag{3}$$

$$\mathrm{P}(\mathbf{I}_t \mid L_{1:t}, F_{1:t}) = \mathrm{P}(\mathbf{I}_t \mid L_t, F_t) \tag{4}$$

The problem stated in Equation 1 can be solved using the recursive Viterbi algorithm [19]. Let us define the maximum probability of observing the images and the trajectory ending up at location k and fall state s at time $t+1$:

$$\Phi_t^{k,s} = \max_{\substack{l_{1:t-1} \\ f_{1:t-1}}} \mathrm{P}(\mathbf{I}, \mathbf{L} = l_{1:t-1}, \mathbf{F} = f_{1:t-1}, L_t = k, F_t = s) \tag{5}$$

If we expand the expression we get:

$$\begin{aligned} \Phi_t^{k,s} =& \mathrm{P}(\mathbf{I_t} \mid L_t = k, F_t = s) \\ & \max_{r,d} \Big\{ \mathrm{P}(F_t = s \mid F_{t-1} = d) \\ & \mathrm{P}(L_t = k \mid L_{t-1} = r, F_{t-1} = d)\, \Phi_{t-1}^{r,d} \Big\} \end{aligned} \tag{6}$$

The fall model is defined as:

$$\mathrm{P}(F_t = k \mid F_{t-1} = i) = \begin{cases} 1 & \text{if } i = 1 \text{ and } k = 1 \\ z & \text{if } i = 0 \text{ and } k = 1 \\ 1 - z & \text{if } i = 0 \text{ and } k = 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

According to the model above, once fallen, the person will remain fallen. z is a small probability that denotes the probability of falling at a given time.

The motion model is defined as:

$$\begin{aligned} &\mathrm{P}(L_t = k \mid L_{t-1} = l, F_{t-1} = f) \\ &\qquad = \begin{cases} \dfrac{1}{Z} e^{-\rho \|l-k\|} & \text{if } f = 0 \text{ and } \|l-k\| < v \\ 1 & \text{if } f = 1 \text{ and } l = k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

The expression above states that if a person has fallen, his location does not change anymore. v defines the upper limit of walking speed, ρ fine tunes the average walking speed.

We assume that all the information is encapsulated in the binary foreground mask, and views are independent given the person's location. For a fixed time t, we write the generative model as:

$$\mathrm{P}(\mathbf{I} \mid L = k, F = f) = \mathrm{P}(\mathbf{B} \mid L = k, F = f) \tag{9}$$

$$= \frac{1}{Z} \prod_c e^{-\Psi(B^c, A^c_{k,f})} \tag{10}$$

where B^c is the binary foreground segmentation obtained from camera c, $A^c_{k,f}$ is the binary image generated by putting the human silhouette in fall state f

at location k and $\Psi(\cdot)$ is a pseudodistance function which is defined for $A, B \in \{0,1\}^{\text{width}\times\text{height}}$ as:

$$\Psi(B, A) = \frac{1}{\sigma} \frac{|B \otimes (1 - A) + (1 - B) \otimes A|}{|A|} \tag{11}$$

An example of real foreground segmentation can be seen in Figure 2.

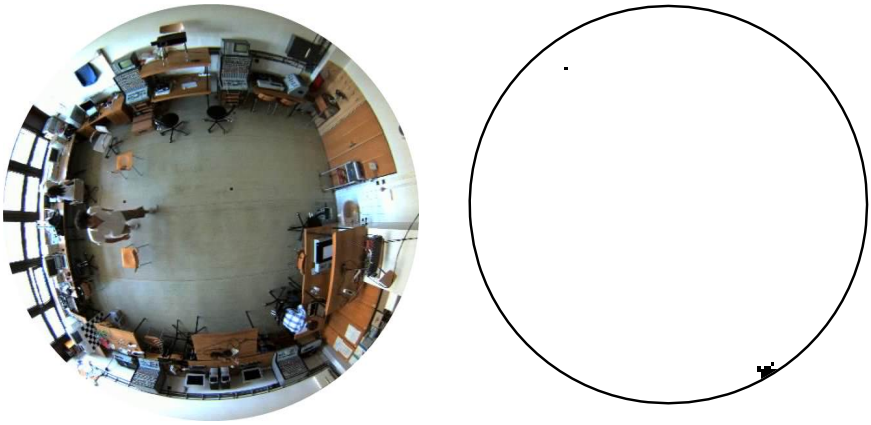

Fig. 2. Frame obtained from top camera and foreground segmentation of the frame

In this paper, human shapes are represented using a cuboid in 3D, instead of a rectangle in the 2D image as proposed in [2]. Example of human silhouettes represented as cuboids can be seen in Figure 3. However, doing this prevents the usage of integral images to speed up computation. Although our approach models a single person and the computations are tractable and fast, in the next section we describe a method to further speed up the pseudo-distance calculation.

5 Fast Integration over Silhouette Region

The method described in the previous section involves evaluation of $\Psi(B^c, A^c_{k,f})$ for all possible locations and states, hence constitutes the bottleneck of the algorithm. Notice that $\Psi(B^c, A^c_{k,f})$ can be expressed as:

$$\Psi(B^c, A^c_{k,f}) = \frac{1}{\sigma} \frac{|B^c| - 2|A^c_{k,f} \otimes B^c| + |A^c_{k,f}|}{|A^c_{k,f}|} \tag{12}$$

If the human silhouettes are straight rectangles, integral images can be used to speed up the computation of $|A^c_{k,f} \otimes B^c|$. However, in our case due to camera distortion and camera positioning human shapes cannot be represented as axis aligned rectangles. We represent human shape as a cuboid and further approximate the

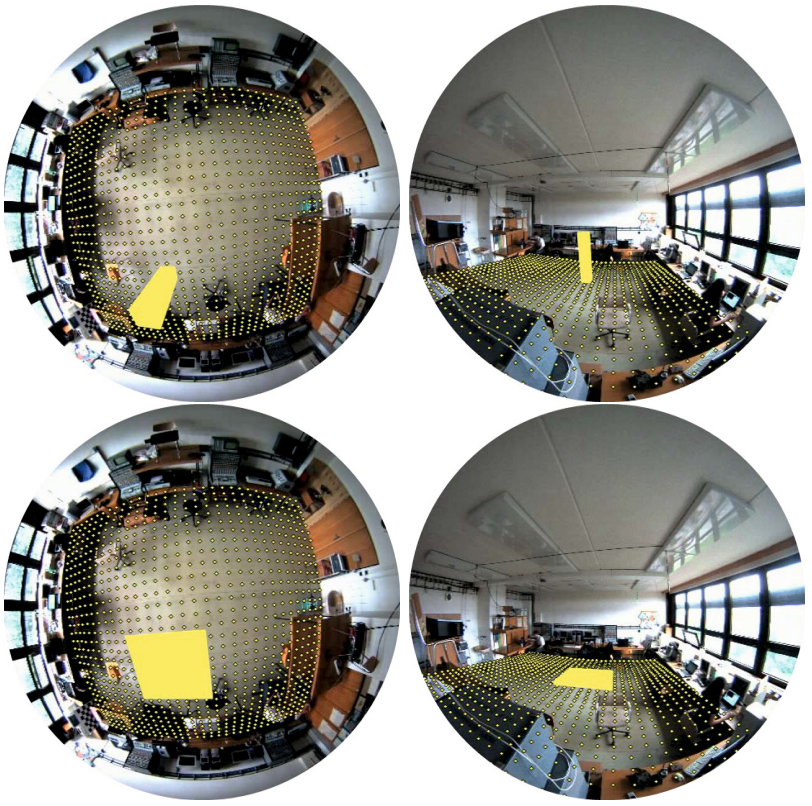

Fig. 3. Locations in the room and the ideal foreground segmentation (human silhouette) for a human at location $k = 545$. Top: $f = 0$, the person is standing. Bottom: $f = 1$ the person has fallen.

silhouettes with multiple axis aligned rectangles; this way psuedodistance calculation can be carried out in constant time. An example of approximating silhouette with rectangles can be seen in Figure 4. Ideal human silhouettes and approximating rectangles need to be generated only once before tracking.

Let S be the region covered by a shape. The goal is to find a set of non-overlapping, axis-aligned rectangles, $\mathcal{R}^* = \{R_i\}_{i=1\cdots N}$, that minimizes non-overlapping area between S and rectangles. More formally:

$$\mathcal{R}^* = \arg\min_{\mathcal{R}} \Omega(S, \mathcal{R}) \tag{13}$$

where

$$\Omega(S, \mathcal{R}) = \left| (S \cup \bigcup_{R_i \in \mathcal{R}} R_i) \setminus (S \cap \bigcup_{R_i \in \mathcal{R}} R_i) \right| \tag{14}$$

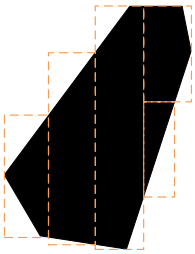

Fig. 4. Representing the human silhouette with five rectangles using the proposed method

Instead of finding the exact solution, we propose a method that iteratively splits rectangles and decreases non-overlapping area. $\mathcal{R}$ is initialized with axis-aligned bounding rectangle of the polygon. At each step, a rectangle with lowest fitness score is removed from $\mathcal{R}$ and split into two. To split a rectangle, the fitness score is calculated for top, bottom, left and right half of the rectangle. Second, the bounding rectangle of the half with lowest score and its complement is added to $\mathcal{R}$.

Fitness score of a rectangle R_i can be defined as inversely proportional to the area of non-overlapping region inside rectangle:

$$f(R_i) = \frac{1}{\Omega(S \cap R_i, \{R_i\})} \tag{15}$$

The fitness score above can be calculated in constant time using integral image of the shape.

We also define a tolerance value to enable early stopping when coverage is good enough. When $f(R_i)$ is greater than tolerance value, the approximation of R_i is considered as acceptable and R_i is not split any further. The number of rectangles covering the polygon can be chosen explicitly, because at each step the number of rectangles is increases by one. The algorithm and an example run can be seen in Figures 5 and 6.

6 Experiments

The proposed fall detection approach was tested on the BOMNI-DB dataset, which involves 5 sets of video pairs obtained using two omnidirectional cameras, containing fainting actions along with other actions of people [20]. Intrinsic and extrinsic camera calibration information is also provided along with the dataset where the camera model described in [21] is used. Each video is approximately 1.5 minutes long (about 750 frames). Videos typically suffer from high noise rates and severe frame loss. Moreover, the person is heavily occluded throughout the video. Keeping these properties of the dataset in mind, the videos can be

```
1: procedure FindCover(shape, maxRectCount, tolerance)
2:     r ← bounding rectangle of shape
3:     R ← {r}
4:     if f(r) > tolerance then
5:         return R
6:     end if
7:     while size(R) < maxRectCount do
8:         r ← arg min_{r∈R} f(r)
9:         R ← R \ r
10:        Calculate f() for top, bottom, left and right half of r
11:        r1 ← bounding rectangle of the half with minimum score
12:        r2 ← other half
13:        R ← R ∪ {r1, r2}
14:        if f(r1) > tolerance and f(r2) > tolerance then
15:            return R
16:        end if
17:    end while
18:    return R
19: end procedure
```

Fig. 5. Proposed algorithm to find rectangles covering a shape

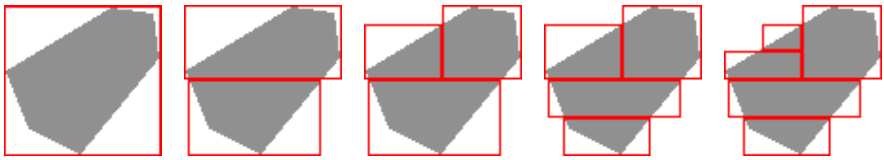

Fig. 6. An example run of decomposing a polygon into five rectangles. At each step, the rectangle with the greatest zero pixel count is selected and split.

considered challenging. When processing video pairs, to handle missing frames, a frame in one video is associated with the closest frame in time in the other video.

The proposed approach is compared with a baseline method (denoted as non-HMM here), which ignores temporal relations and maximizes the term in Eq. 10 for each frame separately.

The foreground detection is performed using the OpenCV 2.4.6 implementation of [22], where intensity value of each pixel is modeled using a mixture of Gaussians. The ground plane is discretized into 961 locations and in the motion model, the transitions from/to the hidden location are allowed only for the locations just in front of the room.

The tracking performance of the system is measured using Multiple Object Tracking Accuracy and Precision (MOTA and MOTP) metrics [23]. The distance between ground truth and prediction is expressed in terms of the overlap of their bounding rectangles. More formally:

$$d(R^{\mathrm{t}}, R^{\mathrm{p}}) = 1 - \frac{|R^{\mathrm{t}} \cap R^{\mathrm{p}}|}{|R^{\mathrm{p}}|} \tag{16}$$

where R^{t} and R^{p} are bounding boxes of the ground truth and prediction, respectively, and $|\cdot|$ is the area operator. The value of the threshold for accepting a prediction is selected as 0.2. Tracking results can be seen in Table 1. Since there is only one object to track, the *mismatch* component of MOTA is irrelevant, and therefore omitted from the table. In the table, *MOTP overlap* values are expressed as overlapping score, i.e. 1 − MOTP.

Table 1. Tracking results on the BOMNI-DB. Improvements in terms of increased MOTA and decreased error over the baseline are indicated in green, whereas the occasional poorer result is shown in red.

#	Camera	MOTP	change	MOTA	change	miss rate	change	false+	change
1	top	0.57	+0.01	0.81	+0.05	0.09	-0.05	0.09	·
	side	0.58	·	0.88	+0.04	0.01	-0.06	0.11	+0.02
2	top	0.60	-0.01	0.94	+0.05	0.00	-0.03	0.05	-0.02
	side	0.53	·	0.94	+0.01	0.01	·	0.05	-0.01
3	top	0.59	·	0.95	+0.01	0.00	-0.01	0.05	·
	side	0.52	+0.01	0.96	+0.10	0.02	-0.09	0.02	-0.01
4	top	0.50	-0.01	0.89	+0.01	0.02	-0.05	0.09	+0.04
	side	0.48	·	0.97	+0.05	0.01	-0.03	0.02	-0.02
5	top	0.53	·	0.99	+0.01	0.00	·	0.00	-0.01
	side	0.47	·	0.98	+0.02	0.00	-0.01	0.02	·
	OVERALL	0.54	·	0.93	+0.04	0.02	-0.04	0.05	·

The performance of the fall detection is measured by false positive and negative counts. False positive is a frame where a fall is reported when there is none. False negative is a fall frame reported as a non-fall. If a false positive or negative is closer than 4 frames (∼half second) to the beginning of the fall event it is ignored, since annotations can be subjective at that scale. Fall detection results can be seen in Table 2. An example fall detection result can be watched at `http://goo.gl/bZYOSw`.

MOTP and MOTA values indicate that the proposed system performs tracking very effectively, with few errors. Using temporal information via hierarchical-HMM improves accuracy compared to the baseline method. The advantage of using HMM becomes more apparent in fall detection, in every video, falls are detected correctly. In case of HMM, false positives and negatives arise near the fall event annotation boundary, which may become subjective at sub-second resolution. However, the baseline approach produces false positives and negatives far from the true fall event.

We have also compared running times using multiple rectangles to represent silhouettes with using silhouette image itself. As it can be seen in Table 3, using multiple rectangles to represent silhouettes provides up to 7 fold speed up.

Table 2. Fall detection results on the BOMNI-DB. FP and FN stands for false positive and negative frame counts respectively.

#	Camera	non-HMM FP	non-HMM FN	HMM FP	HMM FN
1	top	156	20	0	0
	side	138	0	0	0
2	top	3	0	0	0
	side	3	0	0	0
3	top	4	0	0	0
	side	3	2	0	2
4	top	3	0	1	0
	side	1	0	0	0
5	top	1	0	0	0
	side	0	1	0	2

Table 3. Running times of the proposed method using the silhouette image (IMG) and using multiple rectangles (RECT)

#	IMG (sec)	RECT (sec)	speed up
1	340	51	×6.66
2	302	44	×6.86
3	259	38	×6.81
4	253	37	×6.83
6	308	44	×7.00

7 Conclusions and Future Work

In this paper, we have developed a tracking and fall detection system using omnidirectional cameras that is robust to noise, occlusions and missing data. The proposed system merges information provided from all the views and tracks the person. We report our results with a database that contains large amount of missing frames; by fusing multiple camera outputs, we obtain robust performance.

The integral image is a well-known technique for rapidly computing sum of pixel values in rectangular areas repeatedly. Because of the distortion introduced by omnidirectional cameras, they are not directly applicable in our case to rectangular human silhouettes. We have introduced a way to approximate human shapes as collections of non-overlapping rectangles to speed up pseudo distance computation. This scheme can be used to approximate the integral over arbitrary shapes on the image using the conventional integral image, which can be used in any domain where there is a need to compute pixel sums for arbitrary, non-rectangular shapes.

This paper described two methods that constitute useful tools for detecting and tracking people via omnidirectional cameras. As a future work, we plan to extend this framework to more actions, and to multiple subjects in the action scenario. The lack of datasets with detailed annotation is the primary hurdle in this area.

Acknowledgments. This research is supported by Bogazici University projects BAP-6531 and BAP-6754.

References

1. Yasushi, Y.: Omnidirectional sensing and its applications. IEICE Transactions on Information and Systems 82(3), 568–579 (1999)
2. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 267–282 (2008)
3. Roggen, D., Calatroni, A., Rossi, M.: Collecting complex activity datasets in highly rich networked sensor environments. In: Networked Sensing Systems (2010)
4. Behera, A., Hogg, D.C., Cohn, A.G.: Egocentric Activity Monitoring and Recovery. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 519–532. Springer, Heidelberg (2013)
5. Salah, A., Gevers, T., Sebe, N., Vinciarelli, A.: Computer vision for ambient intelligence. Journal of Ambient Intelligence and Smart Environments 3(3), 187–191 (2011)
6. Bourke, A.K., O'Brien, J.V., Lyons, G.M.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait & Posture 26(2), 194–199 (2007)
7. Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laighin, G.O., Rialle, V., Lundy, J.E.: Fall detection-principles and methods. Engineering in Medicine and Biology Society 2007, 1663–1666 (2007)
8. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. Neurocomputing 100, 144–152 (2013)
9. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Fall Detection from Human Shape and Motion History Using Video Surveillance. In: Advanced Information Networking and Applications Workshops, pp. 875–880. IEEE (2007)
10. Nait-Charif, H., McKenna, S.: Activity Summarisation and Fall Detection in a Supportive Home Environment. In: ICPR, vol. 4, pp. 323–326. IEEE (2004)
11. Töreyin, B.U., Dedeoğlu, Y., Çetin, A.E.: HMM based falling person detection using both audio and video. In: Sebe, N., Lew, M., Huang, T.S. (eds.) HCI/ICCV 2005. LNCS, vol. 3766, pp. 211–220. Springer, Heidelberg (2005)
12. Miaou, S.G., Sung, P.H., Huang, C.Y.: A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information. In: Distributed Diagnosis and Home Healthcare, pp. 39–42. IEEE (2006)
13. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. Expert Systems 24(5), 334–345 (2007)
14. Hazelhoff, L., Han, J., de With, P.H.N.: Video-based fall detection in the home using principal component analysis. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2008. LNCS, vol. 5259, pp. 298–309. Springer, Heidelberg (2008)

15. Thome, N., Miguet, S., Ambellouis, S.: A real-time, multiview fall detection system: A LHMM-based approach. Circuits and Systems for Video Technology 18(11), 1522–1532 (2008)
16. Kwolek, B., Kepski, M.: Fall detection using kinect sensor and fall energy image. In: Pan, J.-S., Polycarpou, M.M., Woźniak, M., de Carvalho, A.C.P.L.F., Quintián, H., Corchado, E. (eds.) HAIS 2013. LNCS, vol. 8073, pp. 294–303. Springer, Heidelberg (2013)
17. Nater, F., Grabner, H., Van Gool, L.: Exploiting simple hierarchies for unsupervised human behavior analysis. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2014–2021. IEEE (2010)
18. Huang, Y.C., Rd, C.P., Li, C.: A human fall detection system using an omnidirectional camera in practical environments for health care applications. In: Machine Vision Applications, pp. 455–458 (2009)
19. Forney Jr., G.D.: The Viterbi algorithm. Proceedings of the IEEE 61(3), 268–278 (1973)
20. Demiröz, B., Eroğlu, O., Salah, A., Akarun, L.: Feature-Based Tracking on a Multi-Omnidirectional Camera Dataset. In: ISCCSP (2012)
21. Scaramuzza, D., Martinelli, A., Siegwart, R.: A toolbox for easily calibrating omnidirectional cameras. In: International Conference on Intelligent Robots and Systems, pp. 5695–5701. IEEE (October 2006)
22. KaewTraKulPong, P., Bowden, R.: An improved adaptive background mixture model for real-time tracking with shadow detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, vol. 25, pp. 1–5 (2001)
23. Keni, B., Rainer, S.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing 2008 (2008)