

# ID Management Strategies for Interactive Systems in Multi-Camera Scenarios

Benedikt Gollan

Pervasive Computing Applications

Research Studios Austria

Email: benedikt.gollan@researchstudio.at

Bernhard Wally

Pervasive Computing Applications

Research Studios Austria

Email: bernhard.wally@researchstudio.at

Alois Ferscha

Department of Pervasive Computing

Johannes Kepler University Linz

Email: ferscha@pervasive.jku.at

**Abstract**—This paper presents a novel approach towards identity management strategies for application in large scale interactive systems while separating pure detection from the identity management processes. Detection is achieved by employing a scalable, modal, network-based, real-time multi-camera tracking system in which numerous cameras are used to cover large areas. Objects are detected by employing blob detection and image pre-processing algorithms. The proposed identity handling algorithm covers standard tracking as well as solving Split, Merge, and Handover problems between adjacent cameras. General ID handling strategies are introduced which allow reliable tracking and simple access to high-level movement data for later movement analysis. Results in terms of accuracy and technical feasibility are given, gathered from a pilot installation at a public event.

## I. INTRODUCTION

Interactive systems heavily depend on reliable and extensive information about users to enable meaningful and assisting services. This information can be of any kind, such as gender, age, emotional state, identity, etc. Our research is directed at a large scale system of interactive displays, which is aware of people in their environment, and is able to individually interact with these people based on knowledge gathered from a person's movement and behavior.

For that purpose, a multi-camera-based tracking system, see section II, is employed to gather the movement data which will be used to interactively control the content presented via the displays. Since the system is designed to be portable, a simple installation process and mechanism for calibration of cameras pose further requirements on the camera system. The proposed ID management algorithms are expected to reliably track objects; handle Merge, Split, and Handover situations; and create movement and behavior histories for every person in the scene. These histories, if successfully extracted, could be used to provide interactive control or other offered services.

Note that, in this paper, the term 'identification' will be used in the context of successful multiple target tracking, as opposed to biometric identification (face recognition for instance).

### A. Related Work

Due to the very limited field of view (FoV) of a single camera, multi-camera tracking systems have become the target of research in the last years. Multiple cameras are combined

via network and are either used to obtain multiple views of the same scene, hence bypassing occlusion problems and generating 3D information, or are distributed to cover a large area with single vision perspectives. The latter approaches differ in having overlapping or non-overlapping cameras positions and camera calibration complexity. Cai and Aggarwal [15] used calibrated cameras with overlapping FOVs. The correspondence between objects was established by matching geometric and appearance features of objects in multiple views. Khan et al. [6] used FoV line constraints for tracking in cameras with overlapping views. Englebienne et al. [2] used pairs of cameras to enhance tracking of objects in non-connected cameras using stereo vision. D'Orazio et al. [4] evaluated modifications of Javed's Brightness Transfer Functions [8] for identification via color histogram, whereas Bouchrika et al. [5] used walking patterns as identification criterion. Du and Paiter [12] worked with collaborative particle filters and in their approach, targets are not only tracked in each camera but also in the ground plane by individual particle filters. Mittal et al. [7] presented a system that is capable of tracking numerous people in a cluttered scene using multiple synchronized cameras located far from each other. The views are segmented, taking into account color models for the objects and the background. Approaches employing Bayesian Networks based systems (such as those developed by Chang and Gong [13] and Dockstader and Tekalp [14]) are used for tracking and occlusion reasoning across cameras with overlapping views. Furthermore, overlapping camera FoVs need handover strategies for object transfer between the cameras. Chung et al. [3] and Moller et al. [9] proposed strategies employing multiple cameras with different perspectives. Teixeira et al. [1] started a promising approach with a setting similar to ours, but augmented the visual input with information from localization of cell phones.

The distinct treatment of ID management is also the object of the research by Schumitsch et al. [16] who apply Kalman filters and Bernardin [17] proposed tracking performance metrics for multiple objects. These metrics will be applied for evaluation of our tracking system in section V.

### B. Innovation in This Work

Existing works mainly cover small scale tracking systems of a handful of cameras and do not distinguish between detection and recognition which usually are combined under

the term of 'tracking'. With large scale systems ( $>5$  cameras) demanding sophisticated identification and re-identification algorithms that allow tracking from different perspectives, the need for large scale ID management (IM) arises. Therefore, in this work we focus on handling IDs and present an IM algorithm for identification of people including techniques for Split and Merge handling as well as handling of Handovers between consecutive cameras.

Moreover, real use-case experiments at the 'open day' at University Linz provide accuracy and feasibility results in real conditions. Results of these experiments are presented and discussed in section V.

## II. SYSTEM DESCRIPTION

Figure 1 shows a potential use-case of our interactive system and the basic setup of the Detection Process (DP) and ID management (IM) components. A number of displays are positioned (on exhibition grounds) and are located within the FoV of IP network attached cameras that are mounted top-down from the ceiling.

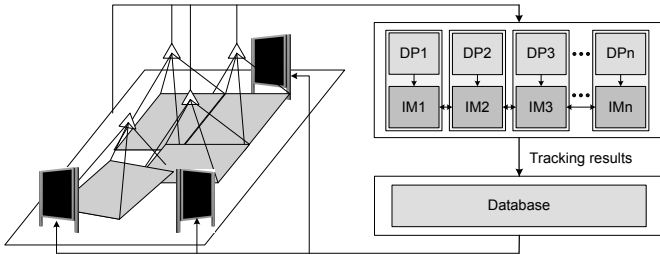


Fig. 1: System architecture including setup of displays and cameras, pairs of Detection Processes (DP) and ID Management (IM) processes per camera and the database for result storage. DP processes are independent whereas IM processes are interconnected.

The top-down camera setting projects all objects into a single plane, thus simplifying detection by minimizing occlusions and simplifying camera calibration. On the other hand, this causes identification on the basis of criteria other than location (color) to be restricted to the limited visible area as viewed from above.

The complete tracking system (see fig.1) has a pair of DP and IM processes per camera, where each DP is running independently per camera and all IM components are closely linked via a network. This provides, theoretically, an arbitrarily scalable system since the number of cameras is not limited by the computation power of a processing unit. Results are stored on a server database which can be addressed by any of the processing units for storing data. The interactive displays receive input for their interaction control. In the current setting, the complete processing and tracking using 4 cameras lead to 40% CPU usage on a Pentium Core2 Duo running at 2.53 GHz. Our expectation is that 6 to 8 cameras can be covered by a single such processor, while larger numbers of cameras would require multiple machines be combined to

share the processing load. This distribution of processing load via network has been successfully implemented and tested on 4 cameras and 2 machines connected via a LAN.

### A. Detection Process

The Detection Process includes image grabbing, image processing, and execution of blob detection for object localization. Therefore, images are captured from the IP cameras (Axis M3203) via video streams in MJPEG format with a resolution of 640x480 pixels at approximately 30 fps. The height of the cameras depends upon the specific use-case, but should be between 3.5 to 6m (as a compromise between resolution and covered area). Image processing is carried out using the openCV [11] libraries, while ffmpeg [10] libraries are used for conversion of the mjpeg-stream. Camera calibration requires a simple manual arrangement of captured images based on small overlaps via a GUI, thus manually assigning positions to the cameras. These positions are used to generate a map of cameras and to transform the local coordinates to global coordinates in the global camera environment.

The detection process will only be described briefly. It includes the steps of smoothing, background/foreground (BG/FG) separation, thresholding, and blob detection. BG/FG separation is computed by comparison of the current frame with a reference frame, which is updated over time to adapt to illumination changes, separation into the single color channels with adaptive thresholding and recombination of the channels with subsequent binarization. The Blob Detection returns the number and attributes as position, size, etc. of the blobs which provides the input to our IM processes.

## III. ID MANAGEMENT - IM

The term ID Management includes every action that involves assignment or re-identification of an ID. This process starts by tracking single objects, and handles Merge, Split, and Handover problems, and ends by writing detection output to the database.

Note that, due to simplification of terms in the following sections, objects will be 'assigned to a camera', means assignment to the processing and identification instance running on behalf of the respective camera. The process cycle for IM components is displayed in Fig. 2.

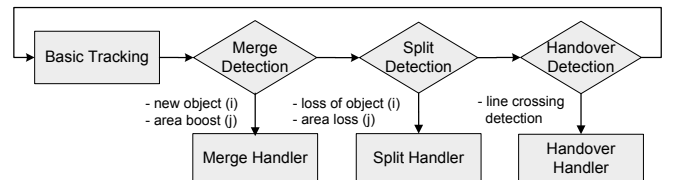


Fig. 2: Architecture of the ID Management cycle including Merge, Split, and Handover Triggers and the respective handlers.

### A. Basic Tracking

Let  $n$  be the number of detected objects  $\Omega$  in the current iteration  $t$ ,  $m$  the number of identified objects  $\Psi$  from the last iteration  $t - 1$  and  $I()$  the assigned identity of the object. All  $\Omega_n$  objects are matched to movement predictions of  $\Psi_m$  objects by minimization of a geometrical error function (eq. 1) which resembles a nearest neighbor optimization. Moreover the minimum distances have to fulfill a maximum constraint of  $\tau$  to trigger successful recognition.

$$I(\Omega_i) = I(\Psi_j) \quad | \quad \min \|\Omega_i, \Psi_j\| \quad (1)$$

and  $\|\Omega_i, \Psi_j\| < \tau$  and  $i \leq n, j \leq m$

In the case of a successful recognition, the localization of  $\Psi_j$  is updated and movement information such as direction vectors, velocity, etc. are computed. If no match for  $\Omega_i$  can be found, a new object  $\Psi_{m+1}$  is generated and parameters (ID, position, boundary box, and covered area) are assigned to  $\Psi_{m+1}$ . This detection cycle is carried out synchronously with image grabbing at approximately 30 Hz. This basic algorithm has proven to successfully track single objects and provides the basis for the following enhancements.

### B. Identification by Color Histogram

In parallel, a color histogram of every  $\Psi_i$  is generated, which is computed on the bounding box of the respective object. This histogram is used for identification of objects in case of Merge and Split situations as described in the next paragraphs. Since the bounding box not only covers object-specific pixels, but includes floor and shadow pixels at the outer regions of the bounding box, the area considered for the histogram is reduced to the central 80% of the box length and width. The histogram is computed for every color layer separately with 8-bit value in a RGB color scheme.

Further, a confidence value is attached to these histograms which controls the update of the respective histogram. Since the histogram becomes more significant and reliable the larger the covered area and the newer the update, this quality criterion is estimated as the area covered by the bounding box degraded by 5% every second. This helps to achieve a histogram that is appropriate for reliable identification. Histogram updates are only carried out in the event of single, non-merged states, and are not shared between cameras (in the case of Handovers) since colors may differ (thus avoiding the need for color calibration or correction for lighting).

### C. Merge Handling

Characteristic for our IM is that an object can carry more than one ID. In case of a merge between two objects  $\Psi_i$  and  $\Psi_j$ , as displayed in Fig. 3, the merged blob is not separated, rather the merged object is assigned the list of all IDs that have taken part in the merging process, and further dealt with in the same way as a single blob.

The Merge detection is triggered by sudden loss of an object and sudden increase in covered area of a nearby object  $\Psi_j$ . If these events are detected at the same time, the Merge Handler

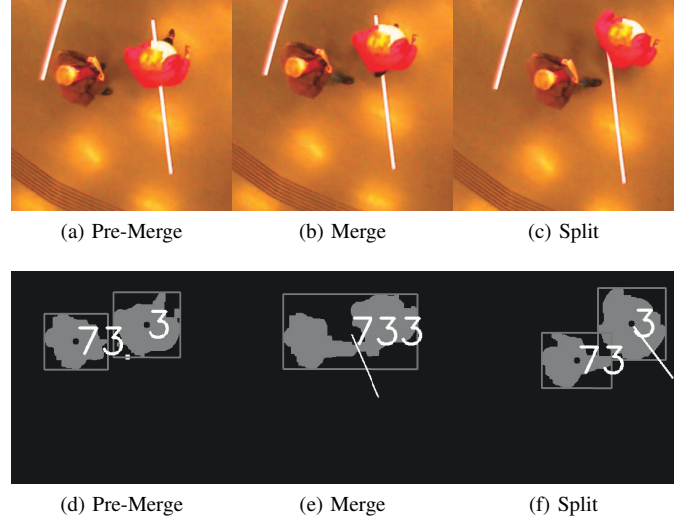


Fig. 3: Tracking visualization for Merge and Split states. Images (a)-(c) show original footage as used for DP, (d)-(f) show blobs with assigned IDs after successful DP and IM. This series presents the process of a typical Merge-Split scenario.

combines the IDs of the two (or more) blobs and assigns these IDs to the resulting object, see Fig. 3.(e).

Thus, an ID pool is generated for a multiple blob containing every potential ID of the affected objects. This comes in handy when dealing with splits in the next section.

### D. Split Handling

In contrast to Merge detection, the Split detection is triggered by sudden appearance of a new object and sudden decrease of the area covered by one of the residual objects  $\Psi_j$ . Further, for positive split triggering, these two objects have to be situated within a certain threshold distance. In case of a positive split detection, the IDs from the splitting object  $\Psi_j$  are split up between the affected objects  $\Psi_j$  and  $\Psi_{m+1}$ .

If the split produces one or more single objects, the respective IDs can be assigned by comparing current histograms taken from the split products to the histogram of the respective IDs from the ID pool. This is achieved by computing the Bhattacharyya distance between affected ID histograms  $H_i$  and  $H_j$  as displayed in (2) with  $N$  being the number of slots for the histogram.

$$d(H_i, H_j) = \sqrt{1 - \frac{1}{\sqrt{H_i H_j N^2}} \sum_I \sqrt{H_i(I) \cdot H_j(I)}} \quad (2)$$

Minimizing the absolute distance error for all possible combinations produces the identification result and assignment of IDs to the single objects is carried out. In case of split products that still have multi-ID state, no definite ID assignment is made and all IDs from the pool are transferred to the new objects. Definite ID assignment is only carried out for single blobs. Detection of single or multi-ID state are made based on object sizes. Since the distance to the objects is constant

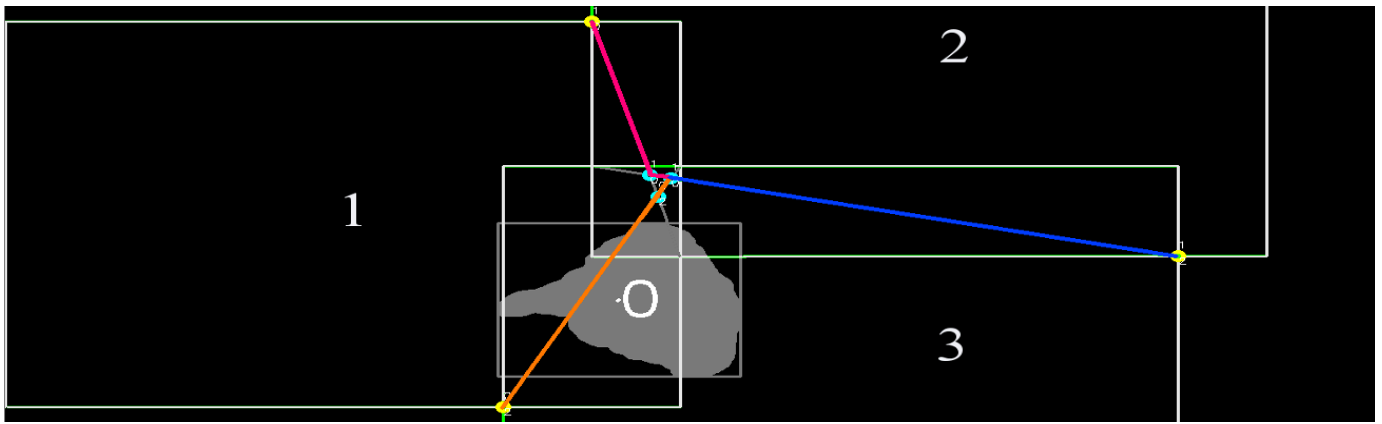


Fig. 4: Handover Lines used for Handover Detection produced by a set of three overlapping cameras. The lines are computed automatically by connecting all intersection points between the image borders and applying a minimum spanning tree.

and overlaps have no crucial influence, the average blob area  $A(\phi)$  is relatively stable, and can be used as decision criterion for the number of persons per object  $N_{\Psi_j}$  (3).

$$N_{\Psi_j} = \lceil \frac{A_{\Psi_j}}{A(\phi)} \rceil \quad (3)$$

However, to better model projection effects, the average object area has to be adapted to the angle  $\phi$  under which the object appears in the camera relative to the camera center.

#### E. Handover Handling

Due to the modal structure of the system an exact separation between neighboring cameras had to be implemented which enables so called ‘Handovers’ of objects between consecutive cameras. This allows explicit assignment of objects to cameras.

This is achieved by creating a dividing line  $l_d$  per overlap in the overlap areas. The final set of dividing lines  $l_f \subset l_d$  is generated by computing the Minimum Spanning Tree of all crossings between  $l_d$  which, even in case of more than two overlapping regions, allows to reduce the separation to a two image problem per line, as displayed in Fig. 4.

A handover is triggered when objects cross the dividing lines  $l_f$  and the called handover handler transfers all relevant information to the responsible camera. Thus multiple detections of objects in overlap regions are evaded by explicit assignment of the object to a camera.

#### IV. REAL USE-CASE EXPERIMENTS

During a public event at the Johannes Kepler University in Linz, footage for evaluation and development of the algorithms was recorded. The public was invited to the university to be informed about research in various presentations. Three cameras were set up in the entrance hall of one of the buildings with small overlaps, covering the main paths between entrance, lift, stairs and information area. The cameras were mounted at a height of 5m and oriented top-down as proposed earlier. In total, about 3000 people were recorded in 5h 30m of video and results excerpted from these recordings are presented in the following.

#### V. RESULTS AND DISCUSSION

In the process of evaluating the proposed tracking system, a segmentation into scenes with high and low density seems obvious due to different performances depending on occupation scales. Low densities scenes are those with 0 to 5 people per camera with a maximum of 11 people in the complete scene of the three cameras, and high densities cover 0 to 9 people per camera with a maximum of 17 people in the complete scene. For both density alternatives, 5000 consecutive frames with altogether 60132 objects have been analyzed and evaluated according to the Multiple Object Tracking Accuracy (MOTA) standard [17] and additional accuracy information, specifically Merge, Split, and Handover statistics, have been computed. The respective results are displayed in table I and II.

TABLE I: MOTA results each on 5000 frames featuring altogether 60132 objects

[%]	miss	mismatch	false positive	MOTA
low density	5.0	3.1	0.2	<b>91.7</b>
high density	15.2	1.6	0.3	<b>82.9</b>

As expected, the MOTA results in table I show different overall performances of our tracking system for different crowd densities. This can be mainly traced back to the increase of missed objects in the high density environment which was due to an underestimation of the number of objects contained in larger, multi-object blobs. The impact of false positive detections can be considered as minor issues, while mismatch rates seem rather small – have to be given special attention, since a mismatch is only counted once in the metric of MOTA, whereas the miss or false positive detection of objects occur not only once, but rather for as long as the object is erroneously detected. Generally, the MOTA metrics have to be treated with care, since miss and false positives have a much larger impact on accuracy results than mismatches, which have a large influence on iterative tracking systems similar to ours.

Additionally, weighting of errors might lead to a more realistic representation of accuracies of tracking systems.

TABLE II: Results [%] on Accuracy of Merges, Splits and Handovers split up by numbers of objects.

merge between	2	3	3	n	overall
low density	95.8	85.7	87.5	-	<b>94.7</b>
high density	89.7	92.0	85.7	91.3	<b>90.4</b>
split between	2	3	3	n	overall
low density	75.4	0.8	-	-	<b>75.8</b>
high density	90.5	71.4	88.9	66.7	<b>81.0</b>
handover of	1	2	3	n	overall
low density	76.2	80.0	-	-	<b>77.4</b>
high density	67.7	75.0	-	-	<b>68.6</b>

In addition to MOTA metrics, in the following, results on the introduced techniques of Merge, Split and Handover procedures are presented. Results on the proposed Merge handling procedure show promising results. Again the lower accuracy rates for the higher density scene mainly stem from an initial underestimation of merged blobs. This phenomenon generally occurs if people enter the scene in groups and hence will be treated as a single object. A possible solution to overcome this issue would e.g. be an estimation of the number of objects inside a blob on the basis of area occupied by the complete blob (as already done during split handling, see 3). This has not been done yet, since the estimation of number of objects per blob according to size is error-prone as an unknown part of the blob area may still be situated outside the scene. For low densities, an estimation of objects per blob could be made as soon as the blob is completely inside the scene.

Split results, as well as results for Handovers, show that the basic procedure is applicable, but still needs some tuning of parameters and refinement of the techniques. For example the movement histories could be used to enhance handling of short Merges and Splits.

Errors in handover handling mainly occur when an object triggers a handover and in the process merges with a different blob which is situated in the destination scene. This will cause an overwriting of the resident object by the new arriving object. Furthermore, hysteresis triggers could be applied for objects which remain in the handover sections for long periods and keep triggering handover procedures.

## VI. CONCLUSION AND OUTLOOK

In this work, we have introduced a novel approach which separates tracking into detection and identification elements and proposed a system of techniques for identity management for large scale multi-camera systems. The need for this separation derives from growing complexity in large-scale multi-camera systems. These techniques are embedded in a completely modal, thus arbitrarily expandable environment, and allow real-time and high level movement analysis for statistical investigations.

Results on real use-case material are presented which do represent lab results, but suggest the applicability of the pro-

posed techniques. The enormous training material captured at the public event at the university will allow further refinement of the tracking procedures and increase in accuracy. Refinement mainly has to be achieved in the interplay between the Merge, Split, and Handover procedures which by themselves show satisfactory results.

## ACKNOWLEDGMENT

This work is supported under the FFG Research Studios Austria program under grant agreement No. 818652 DISPLAYS (Pervasive Display Systems), funded by the Austrian Federal Ministry for Economy, Family and Youth. The Research Studios Austria FG receives the funding for its independent research from the Austrian Federal Ministry for Science and Research.

## REFERENCES

- [1] Teixeira, T., Jung, D., Savvides, A.: Tasking Networked CCTV Cameras and Mobile Phones to Identify and Localize Multiple People Ubicomp '10: Proceedings of the 12th ACM international conference on Ubiquitous computing (2010) 213–222
- [2] G. Englebienne, T. van Oosterhout, B. Krse, Tracking in sparse multi-camera setups using stereo vision, in: Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (2009)
- [3] Chen, C., Yao, Y., Page, D., Abidi, B., Koschan, A., Abidi, M.: Camera handoff with adaptive resource management for multi-camera multi-object tracking, *Image and Vision Computing*, **28** Issue 6 (2010) 851–864
- [4] T. D'Orazio, P.L. Mazzeo, and P. Spagnolo. Color brightness transfer function evaluation for non overlapping multi camera tracking. In Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras (2009)
- [5] Bouchrika, I., Carter, J. and Nixon, M.: Recognizing People in Non-Intersecting Camera Views. International Conference on Imaging for Crime Detection and Prevention (2009)
- [6] Khan, S., Shah, M.: A Multiview Approach To Tracking People In Crowded Scenes Using A Planar Homography Constraint, *ECCV* (2006) 98–109
- [7] Mittal, A., Davis, L.: Unified Multi-Camera Detection and Tracking Using Region-Matching In IEEE Workshop on Multi-Object Tracking (2001)
- [8] Javed, O., Shafique, K., Shah, M.: Appearance modeling for tracking in multiple non-overlapping cameras. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005) 26–33
- [9] Moller, B., Plotz, T., Fink, G.A.: Calibration-free camera hand-over for fast and reliable person tracking in multi-camera setups. 19th International Conference on Pattern Recognition, *ICPR* (2008) 1–4
- [10] Bellard, F.: ffmpeg multimedia system. <http://ffmpeg.sourceforge.net/index.php>
- [11] Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
- [12] Du, W., Piater, J.: Multi-camera people tracking by collaborative particle filters and principal axis-based integration, *ACCV'07: Proceedings of the 8th Asian conference on Computer vision* (2007) 365–374
- [13] Chang, F., Chen, C.J., Lu, C.-J.: A linear-time component-labeling algorithm using contour tracing technique, *Computer Vision and Image Understanding*, **93**, Issue 2, (2004) 206–220
- [14] Dockstader, S.L., Tekalp, A.M.: Multiple camera fusion for multi-object tracking, *Multi-Object Tracking*, 2001. Proceedings. 2001 IEEE Workshop on (2001) 95–102
- [15] Cai, Q., Aggarwal, J.: Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams In 6th International Conference on Computer Vision, (1998) 356–262
- [16] Schumitsch, B., Thrun, S., Guibas, L., Olukotun, K.: The Identity Management Kalman Filter (IMKF), In Proceedings of Robotics: Science and Systems, Philadelphia, PA, USA (2006)
- [17] Bernardin, K., Elbs, A., Stiefelwagen, R.: Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment. In: Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV 2006, Graz, Austria, May 13th (2006)