Employing a RGB-D Sensor for Real-Time Tracking of Humans across Multiple Re-Entries in a Smart Environment

Jungong Han, Eric J. Pauwels, Paul M. de Zeeuw, and Peter H.N. de With, *Fellow, IEEE*

Abstract — *The term smart environment refers to physical spaces equipped with sensors feeding into adaptive algorithms that enable the environment to become sensitive and responsive to the presence and needs of its occupants. People with special needs, such as the elderly or disabled people, stand to benefit most from such environments as they offer sophisticated assistive functionalities supporting independent living and improved safety. In a smart environment, the key issue is to sense the location and identity of its users. In this paper, we intend to tackle the problems of detecting and tracking humans in a realistic home environment by exploiting the complementary nature of (synchronized) color and depth images produced by a low-cost consumer-level RGB-D camera. Our system selectively feeds the complementary data emanating from the two vision sensors to different algorithmic modules which together implement three sequential components: (1) object labeling based on depth data clustering, (2) human re-entry identification based on comparing visual signatures extracted from the color (RGB) information, and (3) human tracking based on the fusion of both depth and RGB data. Experimental results show that this division of labor improves the system's efficiency and classification performance.*[1]

Index Terms — Human re-entry identification, data fusion, visual signature, real-time tracking, opportunistic sensing.

I. INTRODUCTION

The concept of *smart environments* or *ambient intelligence* refers to physical spaces equipped with sensors feeding into adaptive algorithms that enable the environment to become sensitive and responsive to the presence of persons and their individual needs. It is a vision of an imminent future to be brought about by the confluence of consumer electronics, distributed networking and intelligent computing. In a smart environment, people carry out their everyday activities in an easy and comfortable way using information and intelligence that is hidden in the network connecting devices and sensors. Various smart environment components have been implemented and are finding their way to consumers. It is expected that in particular people with special needs such as elderly or disabled people stand to benefit from these developments as such environments offer sophisticated assistive functionalities, supporting independent living and improved safety.

[1] Jungong Han, Eric, J. Pauwels and Paul, M. de Zeeuw are with Centrum Wiskunde & Informatica, Science Park 123, 1098 XG, Amsterdam, The Netherlands (e-mail: {j.han, eric.pauwels, paul.de.zeeuw}@cwi.nl).

Peter H.N. de With is with Eindhoven University of Technology, Den Dolech 2, 5600MB, Eindhoven, The Netherlands.

Contributed Paper
Manuscript received 04/13/12
Current version published 06/22/12
Electronic version published 06/22/12.

0098 3063/12/$20.00 © 2012 IEEE

The core technological problem to be addressed in a smart environment is to detect and track persons, and recognize their actions and intentions. In this paper, we envisage a smart environment (home or office) that attempts to track its occupants during their visits in order to provide appropriate services. Such environments will customarily be equipped with various types of sensors, cameras among them. In a typical scenario (see Fig. 1), a person will enter the environment, pursue different activities during which he or she will usually pop in and out the "observation field" of a number of these sensors, and then take leave. For ease of reference we will call such a visit with intermittent sensor (including camera) observations, an *episode*.

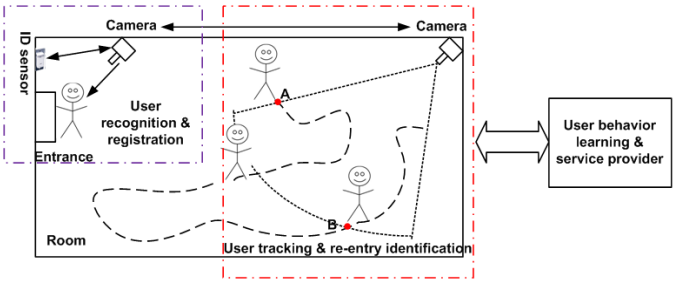

Fig. 1. A typical scenario for a smart environment. The first component is called user recognition & registration (purple block), where the user entering the room is recognized by an ID sensor and his visual signatures are extracted by a camera. The second component is to track the user and to re-identify him when he re-enters the field-of-view of the second camera during his wanderings through the smart environment (events A and B). The last component tries to learn the behavior of users and provide appropriate services.

We contend that in most smart environment scenarios, there typically are two distinct stages to person identification. In the first stage, at the very beginning of an episode as described above, strong identifiers (such as face, iris, fingerprint or electronic ID card) will be used, often to implement strict access control. Such a procedure is quite intrusive and usually requires cooperation of the subject. Once the person has gained access, we enter a second stage in which his or her unique absolute ID is no longer relevant: all we need to know for most practical purposes is that we are observing the same person we have seen earlier. To this end, we will assume that during the initial identification stage (using ID sensor) the system will take the opportunity to also extract salient and easy to spot visual characteristics (such as the color or texture of one's clothing) which will then be used to unobtrusively re-identify the person when he or she re-appears in view. In this paper, we will focus on this second stage and its opportunistic mode of identification based on saliency, as it has the

advantage that there is no need to get a clear view of the person's face in order to re-identify him or her: spotting a salient visual characteristic (such as a red shirt) will suffice. Of course, we need to be aware that saliency will need to be adaptive as it depends on the appearances of all persons to be tracked and therefore requires regular updating.

The task of (re-)identifying persons based on their visual appearance is confounded by the fact that human segmentation and tracking algorithms based on RGB images cannot always provide reliable results. This holds particular true when the environment is cluttered or people suddenly change the illumination conditions, both of which occur frequently in a realistic setting. This paper intends to address this problem by combining two types of cameras. The main feature of our work is that we use complementarily cameras for different algorithmic purposes. Basically, our goal is the implementation of a reliable system for person detection, tracking and re-entry identification based on a consumer sensor that can be used in a smart home environment. The fact that we restrict ourselves to a home or office scenario implies that the number of people to be tracked remains relatively small as it comprises only steady members and some guests.

A. State-of-the-Art of Human Segmentation and Tracking

There has been much development in the field of human detection and tracking in the past years. It can broadly be divided into three categories related to the type of camera used: algorithms using RGB data only, algorithms based on depth data, and algorithms fusing both camera signals.

Human tracking by means of the popular RGB camera has a long history and is still an active field of research. [1-2] provide a broad overview of over one hundred related papers. Here, we only discuss a small number of key techniques. Most human segmentation algorithms start by modeling the background based on a small number of initial frames. Subsequently, a pixel can be labeled either foreground or background depending on the distance of this pixel to a background model at the same location in color/intensity space. Widely used background modeling techniques are the median filter [3] and Gaussian Mixture Models [4]. The basic idea of a tracking algorithm is first to build up an appearance model based on color or/and texture information, which is supposed to be sufficiently distinctive. Next, a matching approach is used to establish the correspondence between the people in successive frames. For instance, mean shift tracker [5] is a real-time non-parametric technique that searches along density gradients to find the peak of probability distributions. The particle-filter technique [6] performs a random search guided by a stochastic motion model to obtain an estimate of the posterior distribution, given an observed appearance model. Recently, several human behavior monitoring systems for consumer applications are implemented based on combining above algorithms. In [7], we present a broadcasting sports analysis system which is intended to be part of a larger consumer media server having retrieval features. The main clue we used for event detection is the moving paths of players that are extracted by a visual tracking algorithm. Our previous work in [8] and the work reported in [9] investigate video surveillance applications for consumer usage, where the former develops a near real-time human posture recognition system for indoor surveillance and the latter intends to improve the multi-object segmentation and tracking algorithms for a varying environment. Generally, both segmentation and tracking algorithms used by above systems rely on the pattern changes of the color/intensity signal at the pixel level. However, this change is *unreliable* in the sense that some unexpected environment changes, such as sudden illumination changes or occlusion, can also trigger the color/intensity change of a pixel. This phenomenon stops such systems from obtaining high accuracy in practical situations.

The systems in the second category utilize a depth camera to detect and track persons. This is a rather new field with little related work. The algorithm [10] first starts a segmentation of the scene in background and foreground (moving) regions using depth information. Tracking is developed based on considering both human motion and depth changes. An alternative method [11] is proposed by Hansen et al., where a background model is built by fusing information from intensity and depth images. The EM (Expectation Maximization) algorithm is used for tracking moving clusters of pixels that are significantly different from the background model. The work done in [12] focuses on improving the foreground detection using graph-cut techniques. In general, the tracking based on depth data will fail in situations where occluded persons have similar depths. Another drawback is that it is impossible to distinguish persons by depth data alone.

The work belonging to the last category fuses the depth information with the RGB information of the image. In [13], one fuses depth and color data to segment the foreground pixels in a video sequence. The basic idea is to generate a sort of probability map for each pixel based on depth data, where a larger probability means that this pixel is likely to be a foreground pixel. Pixels which cannot be unequivocally classified as foreground or background (typically about 1-2% of the image) are re-checked in the color image considering the edge information. Though it conducts a simple object segmentation task, the algorithm executes at a mere 10 frames per second even on a powerful PC. In [14] one fuses laser range and color data to train a robot vision system. For each pixel in the robot's field-of-view, it has color/intensity, depth, and surface normal information, which help to extract 3D features. This technique indeed improves the detection accuracy by 10%, but the speed of the algorithm is far from real-time (a few seconds per image). In [15], two separate particle filter trackers, one using color and the other using depth data, are employed to track objects. The approach is not suitable for real time as it involves heavy processing. Generally speaking, the performance of the algorithm fusing color and depth is better than the algorithm using single type information only. However, the way of fusing data appears too straightforward in the sense that data from different channels are treated equally without considering the specific advantage of each sensor.

B. *Requirements of Home-Used Human Tracking System*

The specific challenges and requirements for a human tracking algorithm that needs to function in a smart home environment system are as follows.

1. The algorithm should be able to track multiple persons, and should be robust against changes in the environment, such as sudden illumination variations or cluttered background.
2. The algorithm should have capability of re-identifying persons who re-enter a room after a short absence. This is essential in order to collect the moving paths generated by the same person.
3. The system should be un-obtrusive, have real-time performance, and preferably use low-cost camera sensors, as it is designed for consumer usage.

To address the first requirement, we exploit the benefit of the integrated depth camera for person segmentation to avoid having to use complicated background modeling techniques. In contrast with the color/intensity information, the depth of an object is insensitive to environmental changes. Consequently, background subtraction based on depth data enables to segment moving humans in most practical situations. In order to track multiple persons accurately, we fuse the color and depth of an object in a probabilistic fashion. By doing so, we can even handle complicated situations where two persons wearing similar clothing are partially occluding each other.

With respect to the second requirement, we distinguish between different persons by employing a color histogram incorporating both textural and spatial information derived from the appearance. Here, the color and texture of a human are obtained from the RGB camera. The extraction of human body parts (spatial information) is facilitated by human segmentation results.

Regarding to the last requirement, we use different camera sensors for different algorithmic modules, rather than running the same algorithm for each camera and generating an outcome based on results from two channels. Furthermore, we choose a low-cost RGB-D sensor for our implementation. Experiments (reported in Section IV) will show that the quality is sufficient for our application.

In the sequel, we first present a system overview with a task graph in Section II and then describe in detail our key techniques in Section III. Experimental results are provided in Section IV. Finally, Section V draws conclusions.

II. Overview of Proposed System

Fig. 2 depicts our system architecture with its main functional units and data flows. The functions of the key modules are as follows.

- *Object labeling.* This module takes the depth images as the input, and outputs the location(s) of detected object(s). All detected objects are put into a waiting list which will be checked by our human detection module.
- *Human Detection.* This module scans all detected objects and "promotes" an object to *person* status if it passes the evaluation.

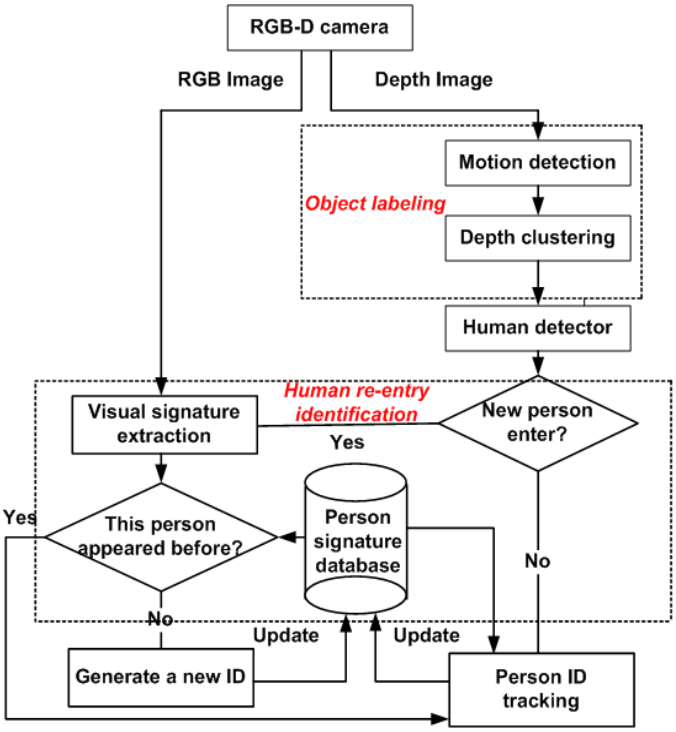

Fig. 2. System overview.

- *Human re-entry identification.* Once we have detected a new person entering, we extract visual signatures for this person from the RGB image. The signatures of this new person are compared to the signatures already stored in a *person* database. If it matches with the signature of an existing person, we assign the ID of the existing person to this *new* person, because it must be the case that he or she has returned after a short leave. If we cannot find a good match, we assign a new ID for this new person, and will keep tracking this ID in the successive frames.
- *People ID tracking.* We track all active (i.e. observed) people at every frame based on taking both depth continuity and RGB appearance similarity.

III. Techniques for Proposed System

We start by introducing the property of depth camera and RGB camera, where we show how these two cameras complement to one another during a human tracking. Subsequently, we provide more details for each technical module mentioned in the system overview.

A. *The Properties of Depth Camera and RGB Camera*

A depth camera is also called *Time-of-Flight* sensor, as it uses the time difference between pulsed infrared light and its reflection on objects in the scene to provide a dense depth map for the scene at high frame rates. Based on these depth data, it is feasible to generate a faithful background model for the scene, which can then be used to detect foreground objects. In turn, this allows us to generate easy and reliable segmentations of the humans in the observed realistic environment. Due to the nature of the depth map, this model is not sensitive to

changes in illumination (e.g. from shadows) or lack of contrast, all of which are well-known problems vexing traditional tracking algorithms based on RGB images.

However, a depth sensor comes with its own limitations. Most notably, as appearance information is completely discarded, tagging an object based on its visual features is impossible. As a consequence, although it is relatively easy to detect humans in a scene, it is impossible to re-identify them when they return into the sensors field of view after a brief absence. Another drawback is that segmentation relying on depth data may spuriously lump together two distinct objects when they are touching (e.g. a person and the chair he's sitting on). This is due to the fact that depth segmentation assumes that the depth of a single object varies smoothly, and vice versa. A RGB camera, on the other hand, provides color and intensity information of an object, which are important for distinguishing objects and visualizing the scene.

Traditional approaches for detecting and tracking humans in a home environment utilize RGB cameras only. They work well under the standard assumptions, such as an uncluttered background, constant illumination, and high contrast between foreground (persons) and background. Unfortunately, those assumptions seldom apply in realistic settings.

From the above discussion, it transpires that the depth camera and RGB camera actually complement one another. It makes sense to combine the data from both cameras in heterogeneous instead of a homogeneous fashion, resulting in a more robust human tracking and re-entry identification system.

B. *Object Labeling by Using Depth Data*

There are two steps, where the first one is the motion detection and the second step is called depth clustering. We employ a background subtraction algorithm to detect the moving pixels in the depth image, which can be formulated as:

$$M(x,y) = \begin{cases} 1, & if\ |D(x,y) - B(x,y)| \geq T \\ 0, & otherwise \end{cases}. \qquad (1)$$

Here, (x, y) is the coordinate of a pixel, $D(x, y)$ represents the depth value of this pixel and $B(x, y)$ refers to the depth of the background. If the difference of $D(x, y)$ and $B(x, y)$ is larger than a threshold *T*, the pixel is labeled as a moving pixel. To simplify the algorithm, we always use the previous depth image as the background of the current depth image.

Moving pixels detected in the previous step are seeds to initialize a clustering algorithm, which is the second step of our object segmentation. This clustering algorithm checks the *depth* continuity of neighboring pixels of the seeds. Finally, the algorithm returns with several separated clusters, each corresponding to a different object. In general, the algorithm is quite simple and fast, and enjoys good performance.

C. *Detecting Humans*

The task of this module is to determine whether or not a newly detected object is human-like. If affirmative, we extract a human visual signature (based on features) and compare it against a database of recorded visual signatures of previously observed persons. This is also responsible for deciding a moment when we should compare this detected person with the visual signature in the database. In principle, we can detect a person at any postural moments. But the problem is that the visual appearance of a person is highly dependent on his posture: a person can be standing, sitting or even squatting and his visual appearance will vary accordingly. To solve this problem, we defer the computation of visual characteristics until we are reasonably sure that he is standing. By doing so, we can ensure that we always extract comparable visual signatures, resulting in accurate person re-identification.

The human detection algorithm relies on two parameters, the first one of which is called the *stability* of the object, while the second parameter is the *height* of a moving object. Basically, a moving object can be promoted to be a human only when it is *stable* with *sufficient height*. To measure the stability of an object, we check the changes of the object size in successive five frames. More specifically, we keep the size of an object for five successive frames, and compute the size change of this object between each frame and the next. If all four size changes are less than 10% of the object size, we consider that object as stable. To measure the height of an object, we use the length of the object in the image domain. However, this length is varying in terms of the distance of the object to the camera. In [16], it is stated that the relationship between the distance to the camera and the object length is linear for small look-down angles. This relationship between the object length l and the distance d can be defined as:

$$l(d) = a_1 d + a_2, \qquad (2)$$

where a_1 and a_2 are parameters which can be derived from measurements. To this end, an off-line calibration procedure is required, where a person of known height walks through the scene at random. At each instance, the length of this person can be computed from the binary map generated by background subtraction algorithm, and the depth information of this person can be directly obtained from the depth image. Afterwards, the parameters can be estimated using least squares techniques. Once we have obtained this relationship, we can compute the length of the human, who was involved in the off-line calibration, at any position. Furthermore, we define the lowest and the tallest height of persons accepted by our human detection, which are $150\,cm$ and $200\,cm$, respectively. Given the physical height of the human, who is involved in the calibration, a ratio between accepted height and calibrated height in the physical space can be easily computed. This ratio equals to the ratio in the image domain when camera look-down angle is small. Therefore, we can compute the accepted length of a human, given the depth information. In other words, we can use the length of a moving object at certain position to decide whether or not its height is reasonable for a human.

Determining a human based on the stability and height is not an optimal solution in the general situation. However, it

works well in a home environment, because it rarely happens that an inanimate object with human height is moving in the room. In the end, the simplicity of this algorithm helps us to establish a real-time system.

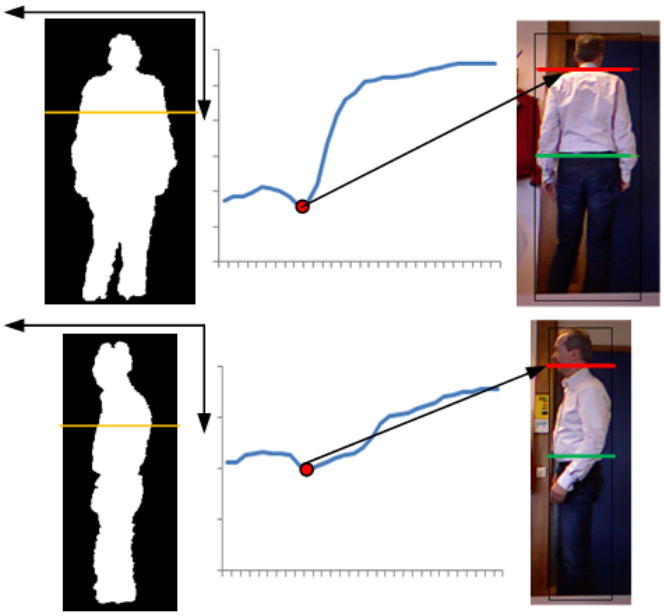

Fig. 3. Human body parts detection. Left: human silhouette. Middle: curve for projected histogram, where the red dot indicates the position of local minimum. Right: detection result.

D. Human Re-Entry Identification

This module is responsible for re-identifying a person when he returns. For the applications we have in mind, we need to be able to track persons across successive appearances in the scene and tag them with a persistent ID label. Most existing human tracking systems for surveillance applications, such as the systems we developed in the previous work [8][9], do not have this function. They simply assign an ID whenever a person enters the scene, and track this ID until he disappears from view. If he happens to return, a new ID will be assigned. Obviously this is unacceptable in, say, a smart home application where the tracking functionality would be relied upon to offer person-based services.

To enable appearance-based matching between successive occurrences of the same person, we use extended color histograms. More precisely, our histogram includes not only color information but also texture information of the pixels, comparing with traditional color histogram or correlograms [17]. In more detail, we start by detecting the head region of a person based on a fast inspection of the shape of the human silhouette. Once the head region of a person is extracted, it is easy to locate the torso and leg region as well, since humans have fixed length ratios between different body parts. The extraction of the head region is conducted on a human silhouette map already generated by our object labeling algorithm. We first obtain a searching area, which is the top 1/3 part of the entire human body. Next, we project the human silhouette along the horizontal direction, and trace the silhouette's width. We illustrate two typical examples in Fig. 3, where the first one is a frontal view of a human, whereas the second one depicts its profile. Since the width of the neck is less than that of the flanking head and shoulders, one can retrieve the position of the neck by locating the corresponding local minimum in the width curve. Afterwards, the torso region as well as the leg region can be deduced by using the human body ratio. The left pictures in Fig. 3 show the body parts detected by our algorithm for both cases.

Up to this stage, we have divided the entire human body into three parts: head, torso and legs. For the purpose of distinguishing different persons based on their appearance (other than face recognition), the information of the head region may not be very important, because the color of this region (face and hair) shows little difference between different persons from the *camera* point of view. Therefore, our algorithm does not extract visual signatures from this region. Instead, we extract visual signatures from torso and leg regions, because it rarely happens that two persons are wearing the clothing with the same colors. As we mentioned before, we make use of color histogram to describe the human appearance. However, we consider that the texture information is also important. Therefore, our histogram also encodes the texture information of pixels involved in the computation. Let $\{x_i\}_{i=1\ldots n}$ be the pixel locations in the defined region, such as torso region. The function b: $R^2 \rightarrow \{1\ldots\mathrm{m}\}$ associates to the pixel at location x_i the index $b(x_i)$ of its bin in the quantized feature space (RGB space in our case). The probability of the feature $u = 1 \ldots m$ is then computed as

$$q_u = C \sum_i^n w_i \delta[b(x_i) - u], \tag{3}$$

where δ is the Kronecker delta function. The normalization constant C is derived by imposing the condition $\sum_{u=1}^{m} q_u = 1$. Parameter w_i measures the texture intensity of the pixel, which is defined as

$$w_i = \begin{cases} 1, & if\ x_i\ is\ an\ edge\ pixel \\ 0.5, & otherwise \end{cases}. \tag{4}$$

Here, we use the Canny operator to detect the edge pixel. We apply the same histogram computation introduced above to the torso region and the leg region separately. Afterwards, we concatenate both histograms into the final histogram for that person. When comparing two histograms, we compute the angle between two histograms, defined as

$$\alpha = \arccos\left(\frac{\sum_i h_1(i)h_2(i)}{\sqrt{\sum_i\left(h_1^2(i) + h_2^2(i)\right)}}\right), \tag{5}$$

where $h_1(i)$ and $h_2(i)$ are appearance histograms of two persons. In our algorithm, if the angle between two histograms is smaller than a threshold, we conclude that they belong to the same person; otherwise, they are labeled as different persons.

E. Human ID Tracking

The goal of the human ID tracking algorithm is to find the new location of an *activated* person in the current frame, and further update its information encoded in the ID. The *activated* ID means that the person has appeared and was already labeled in the previous frame. For instance, suppose we have detected five persons in the current frame, and we also know there are five activated IDs at this moment. The task of the ID tracking algorithm is to find out who corresponds to which ID in the current frame.

In order to determine this correspondence, our algorithm employs a probabilistic framework in which both the continuity in the change of depth as well as appearance similarity are considered. In other words, we compound the information from both depth camera and RGB camera, leading to a better decision making. We notice that the depth of an object should not change dramatically between two successive frames, and the change usually follows a *trend* which can be estimated based on the changes in the previous frames. For example, a person is moving in the room with a constant speed. If we check the depth of this person over time, the changes of depths between each two successive frames are more or less the same, as the person has constant speed. In other words, the bigger depth difference between an active ID (always keeps the depth of the person in the last frame) and one candidate detected in the current frame, the smaller probability of the candidate corresponding to that ID. Moreover, the appearance of a detected people is expected to be similar to that of the human ID if they are corresponding to each other. Let us now discuss how we can compute the correspondence probability. Assume that the existing human ID is T_i and D_j denotes the j th candidate extracted in the current frame. We assume that the depth change follows Gaussian distribution, and compute the average depth change (μ_{dc}) and its variance (σ_{dc}) for T_i based on its last 10 frames. If we only consider the depth change continuity, the probability of T_i matching with D_j, given the depth change d_c, can be estimated by:

$$p_d(T_i \to D_j) = \frac{1}{\sqrt{2\pi\sigma_{dc}^2}} e^{-\frac{(d_c-\mu_{dc})^2}{2\sigma^2}}. \quad (6)$$

Assume the appearance histogram of T_i is h_T, and h_D denotes the appearance histogram of D_j. If we only consider the appearance similarity, the probability of T_i matching with D_j can be computed by (using the Bhattacharyya distance):

$$p_a(T_i \to D_j) = \sum_{u=1}^{m} \sqrt{h_T(u)h_D(u)}. \quad (7)$$

Since the depth continuity and the appearance similarity are equally important for our decision, finally the probability of T_i matching with D_j is a linear combination of $p_d(T_i \to D_j)$ and $p_a(T_i \to D_j)$, which is

$$p(T_i \to D_j) = 0.5p_d(T_i \to D_j) + 0.5p_a(T_i \to D_j). \quad (8)$$

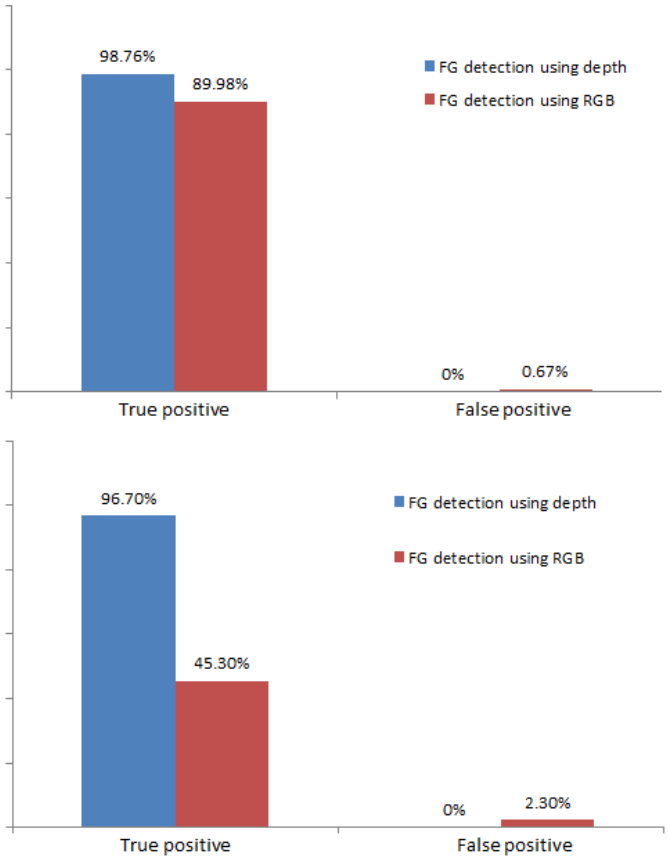

Fig. 4. True positive and false positive of foreground pixels detected by using depth data and RGB data respectively. Top: the lighting condition is stable and the foreground is different with the background. Bottom: the light is stable but the foreground and background are similar.

IV. Experimental Results

Our proposed system is implemented in C++ on a Laptop PC platform (Dual core 2.53 GHz, 4 GB RAM) with a 64-bits operation system. Our software relies on OpenNI library and OpenCV library, where the former provides the functions to drive the RGB-D sensor and implements the object labeling module, while the second library provides basic computer vision algorithms. Below, we describe results for separate parts of our system and the efficiency of the system as a whole.

A. Object Labeling Evaluation

Our object labeling module has been evaluated and compared with a GMM-based foreground pixel detection algorithm. To highlight the difference, we test algorithms in three different situations. First, a person is moving in a room with uniform and stable lighting conditions and his cloth is clearly different with the background. In the second situation, the moving person wears clothing, which happens to be similar to the background. In the last situation, a person suddenly turns off the light in the room. Obviously, the last two situations are more challenging, but they occur regularly in realistic environments. Fig. 4 gives comparison results (the ground truth is generated manually) for the first situation and the second situation as well. The left bars represent the percentages of true positives of the detections, and the right bars indicate the percentages of false positives of the detections. It can be noticed from the results that GMM

algorithm based on RGB images performs properly in the first situation. However, the performance drops significantly when dealing with the second situation. For the third situation, we do not provide quantitative comparison, because the performance difference is extreme. In Fig. 5, we show two examples, where the foreground and the background are quite similar in the left example, and the illumination of lighting has a sudden change in the right example.

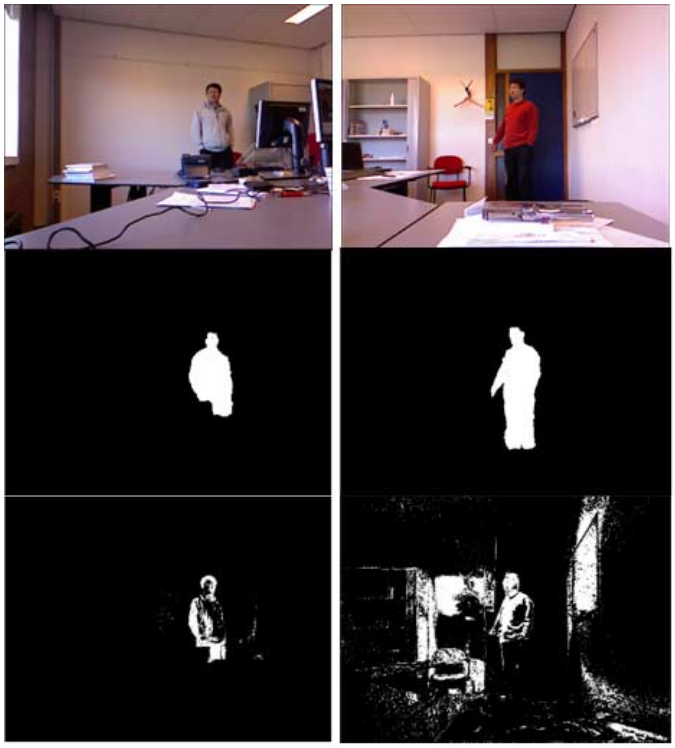

Fig. 5. FG detection results. From the top to the bottom, RGB images from The RGB-D sensor, FG detection results using depth data and FG detection results using RGB data, respectively. The left example shows the situation of clothing hard to distinguish from the background, and the right one reports the results when the person suddenly turns on the light.

We have also evaluated our head region detection algorithm based on 2000 frames captured at different locations, because its accuracy directly influences the visual signature extraction. Our evaluation criterion is that the bottom line of the rectangular box should be around the person neck region. The result reveals (96.1% accuracy) that our algorithm is very accurate, because the head region is not precisely detected only on 78 frames. Fig. 6 illustrates several successful examples in extremely difficult cases, where persons were occluded by furniture or by other person.

B. Human Re-Entry Detection and ID Tracking Evaluation

To evaluate our person re-entry identification module, we have positioned the RGB-D sensor in a living room for 30 minutes, and asked persons to leave and come back for 35 times. During the test, persons used 5 different coats. The algorithm mistakenly assigns IDs on 8 occasions. Most failures are explained by persons wearing coats with similar colors. The remaining failures are due to the fact that human posture when registering in the database differs with the posture when the person enters again.

Fig. 6. Head region detection at different locations, where red line indicates the neck position.

We have also evaluated the human tracking module based on 5 videos (in total 2600 frames) involving 2 to 3 persons. We have compared it with the mean shift tracker [5] and also a particle filter tracker [6]. The evaluation criterion is that the bounding box should include at least 70% of the human silhouette. To test the robustness of our algorithm, we change the illumination of the lighting during one video. The overall accuracy of our tracker is 96.27%, the accuracy of the particle filter tracker is 83.54%, and the accuracy of the mean shift tracker is 71.23%. According to the results, the mean shift tracker does not handle occlusion properly, and the particle tracker deteriorates on the video when we change the illumination. In contrast to these two algorithms, our tracker fails only once when two person pass by each other (occlusion) after changing the lighting conditions. Our tracker actually works properly even after changing the lighting. The failure reason is that we need to re-identify one person after he was occluded, but the visual signature of this person was generated prior to changing the lighting. This illumination change confuses our re-identification module. We can solve this problem if we update the visual signature of a person regularly. In Fig. 7, we demonstrate two examples, where the first one shows that we still track persons after an illumination change, while the second example visualizes that we can track 3 persons (2 adults and one kid) simultaneously. In the example of 3 persons, we label the kid as an “Anonym”, because he does not pass the evaluation for human height in this case. However, we still track his movements over time.

C. System Efficiency

Finally, we have also measured the execution time of the entire system, because the efficiency of the system is important for smart environment applications. We measure it for the cases where the number of moving persons is varying. For each case, we have computed the average time-consumption based on 100 frames. For the 1-person case, the entire system costs 41.3 ms per frame to handle two channels of signals. For the 2-person case, the time consumption is around 73.8 ms per frame on the average. For the 3-person

case, the average running cost is 97.1 ms per frame. Our algorithm still can process two channels with 10 fps, even when there are multiple persons in the scene.

Fig. 7. Examples for human tracking. Top: we change the illumination. Bottom: we track 3 persons.

V. CONCLUSION

We have proposed a two-camera system based on a RGB-D sensor, which enables person detection, tracking and re-entry identification. The system can be the stepping stone for smart environment applications, where sensing user's location and behavior is essential. We intend to use cameras generating complementary data for different algorithmic purposes and exploit their different properties and specific advantages. By doing so, our system can achieve real-time performance with sufficient accuracy. The results testing at different locations show that the accuracy of object labeling is about 95% in a realistic environment. We can successfully re-identify persons leaving and returning to a room in 80% of the cases. The tracker based on fusing images from two channels achieves an accuracy rate of about 96% in occlusion and illumination change cases, which outperforms other existing algorithms. As our system is efficient and fast, it enables a realistic implementation of a smart environment system.

We are still improving the human detector module to execute with a more general descriptor for the human shape instead of relying on general height information of a human. We will report on this in a forthcoming paper.

REFERENCES

[1] J. Aggarwal, and Q. Cai, "Human motion analysis: a review," *Proc. IEEE Workshop on Nonrigid and Articulated Motion,* pp. 90-102, Jun. 1997.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys,* vol. 38, no. 4, Article 13, Dec. 2006.

[3] R. Culter, and L. Davis, "View-based detection," *Proc. ICPR,* vol. 1, pp. 495-500, Aug. 1998.

[4] Z. Zivkovic, "Improved adaptive Gaussian Mixture Model for background subtraction," *Proc. ICPR*, pp. 28-31, Aug. 2004.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 25, no. 5, pp. 564-577, 2003.

[6] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comp.,* vol. 21, no.1, pp. 99-110, Jan. 2003.

[7] J. Han, D. Farin, P. de With and W. Lao, "Real-time video content analysis tool for consumer media storage system," *IEEE Trans. Consumer Electron.,* vol. 52, no. 3, pp. 870-878, Aug. 2006.

[8] W. Lao, J. Han, and P. de With, "Automatic video-based human motion analyzer for consumer surveillance system," *IEEE Trans. Consumer Electron.,* vol. 55, no. 2, pp. 591-598, May 2009.

[9] J. Kim, D. Yeom and Y. Joo, "Fast and robust algorithm for tracking multiple moving objects for intelligent video surveillance systems," *IEEE Trans. Consumer Electron.,* vol. 57, no. 3, pp. 1165-1170, Aug. 2011.

[10] A. Bevilacqua, L. Di and S. Azzari, "People tracking using a Time-of-Flight depth sensor," *Proc. IEEE Int. Conf. Video and Signal based Surveillance*, pp. 89-93, Dec. 2006.

[11] D. Hansen, M. Hansen, M. Kirschmeyer, R. Larsen and D. Silvestre, "Cluster tracking with Time-of-Flight cameras," *Proc. CVPR workshop on TOF-CV*, June 2008.

[12] O. Arif, W. Daley, P. Vela, J. Teizer and J. Stewart, "Visual tracking and segmentation using Time-of-Flight sensor," *Proc. ICIP*, pp. 2241-2244, Sept. 2010.

[13] R. Crabb, C. Tracey, A. Puranik and J. Davis, "Real-time foreground segmentation via range and color imaging", *Proc. CVPR Workshop on TOF-CV*, June 2008.

[14] S. Gould, P. Baumstarck, P. Quigley, M. Ng, and A. Koller, "Integrating visual and range data for robotic object detection," *Proc. ECCV Workshop on Multi-camera and Multimodal Sensor Fusion*, June 2008.

[15] L. Sabeti, E. Parvizi, and Q. Wu, "Visual tracking using color cameras and Time-of-Flight range imaging sensors," *Journal of multimedia*, vol. 3, no. 2, pp. 28-36, June 2008.

[16] P. Remagnino, A. Shihab, and G. Jones, "Distributed intelligence for multi-camera visual surveillance," *Pattern Recognition*, vol. 37, no. 4, pp. 675-689, Apri. 2004.

[17] J. Huang, S. Kumar, M. Mitra, W. Zhu and R. Zabih, "Image index using color correlograms," *Proc. CVPR*, pp. 762-768, June 1997.

BIOGRAPHIES

Jungong Han received his Ph.D. diploma in communication and information system from XiDian University, China, in 2004. In 2003, he has been a visiting scholar at Internet Media group of Microsoft Research Asia, China, working on scalable video coding. In December of 2004, he joined the department of Signal Processing Systems (SPS) at the Technology University of Eindhoven (TU/e), The Netherlands. From 2005 to 2010, he was leading the research branch for video content analysis within SPS-VCA group. In December of 2010, he joined the Multi-agent and Adaptive Computation research group at the Centre for Mathematics and Computer Science (CWI) in Amsterdam, where he is currently participating in an EU-FP7 project for sensor network. His research interests are content-based video analysis, human behavior analysis and multi-sensor data fusion. He has written and co-authored over 50 papers including 3 invited papers in these areas. He served as the associate editor, TPC member, guest editor, session chair, and the reviewer for several international conferences and journals.

Eric J. Pauwels joined the computer vision research group at ESAT (Leuven University, Belgium) after completing his PhD in Mathematics, and worked on various mathematical problems in computer vision, including differential, semi-differential and algebraic invariants and their application to object recognition. In 1999, Dr. Pauwels joined the Signals and Images research group at the Centre for Mathematics and Computer Science (CWI) in Amsterdam where he focuses on two topics: content-based image retrieval, and multimodal camera and sensor networks for situational awareness in smart environments. Dr. Pauwels has contributed to numerous national and European projects and was the scientific coordinator of the FP6 Network of Excellence on Multimedia Understanding through Semantics, Computations and Learning (MUSCLE). He founded and acted as the first chairman for the ERCIM Working Group on Image and Video Understanding. He also organized and chaired the first international workshop on Distributed Sensing and Collective Intelligence in Biodiversity Monitoring.

Paul M. de Zeeuw is a numerical mathematician, affiliated at the Centrum Wiskunde & Informatica, Amsterdam (NL), since 1979. He studied mathematics and computer science at the University of Leiden and obtained his PhD thesis from the University of Amsterdam. He authored and co-authored many papers on multigrid algorithms for the solution of partial differential equations. One paper in particular is much cited and the accompanying computer code is widely used. De Zeeuw has also been participating in image processing projects, as a spin-off thereof two Matlab toolboxes have been built and made available on the web. Further, he has been author at the Dutch Open University on the topic of numerical linear algebra, and was the secretary of the Dutch-Flemish Numerical Analysis Society from 1997 till 2002, including being editor of its newsletter. He has acted as a reviewer of project proposals. Present focal points are applications of multi-resolution methods in image processing, including image fusion and content-based image retrieval.

Peter H.N. de With graduated in Electrical Engineering from the University of Technology in Eindhoven. In 1992, he received his Ph.D. degree from the University of Technology Delft, The Netherlands. He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department and set-up the first DCT-based compression systems. From 1985 to 1993, he was involved in several European research projects on SDTV and HDTV recording. He was the leading video compression expert for the DV camcorder standard from 1989-1993. In 1994, he became a member of the TV Systems group at Philips Research Eindhoven, where he was leading the design of advanced programmable video architectures and a senior TV systems architect. In 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty Computer Engineering and heading the chair on Digital Circuitry and Simulation. Between 2000 and 2007, he was with LogicaCMG in Eindhoven as a principal consultant and distinguished business consultant and simultaneously, he is professor at the University of Technology Eindhoven, at the faculty of Electrical Engineering, heading the chair on Video Coding and Architectures as part of the Dept. on Signal Processing Systems. In the period 2008 – 2010, he was vice president Video Technology at CycloMedia Technology, Waardenburg, The Netherlands, establishing image analysis applications. Early 2011, he was appointed scientific director of the Centre for Care & Cure Technology at the University of Technology Eindhoven and theme leader on smart diagnosis for the University. Mr. De With is Fellow of the IEEE and co-author over 50 refereed international book chapters and journal papers and over 250 international conference papers, and holding over 40 international patents. He is a co-recipient of multiple paper awards like the IEEE CES Transactions Paper Award (several), VCIP and ICCE Best Paper Awards and Invention awards. He is a program committee member of the IEEE CES, ICIP and SPIE VCIP and chairman or board member of various international working groups and foundations.