FUSION OF MULTI-MODAL SENSORS IN A VOXEL OCCUPANCY GRID FOR TRACKING AND BEHAVIOUR ANALYSIS

Martin Hofmann[1], *Moritz Kaiser*[1], *Hadi Aliakbarpour*[2], *Gerhard Rigoll*[1]

[1] Technische Universität München, Institute for Human-Machine Communication, Munich, Germany
[2] University of Coimbra, Institute of Systems and Robotics, Coimbra, Portugal

ABSTRACT

In this paper, we present a multi-modal fusion scheme for tracking and behavior analysis in Smart Home environments. This is applied to tracking multiple people and detecting their behavior. To this end, information from multiple heterogeneous sensors (visual color sensor, thermal sensor, infrared sensor and photonic mixer devices) is combined in a common 3D voxel occupancy grid. Graph cuts are used for data fusion and to accurately reconstruct people in the scene. A Viterbi tracking framework is applied to track all people and simultaneously determine their behaviour. We evaluate the proposed fusion scheme on the PROMETHEUS Smart Home database and show the impact of different sensors and modalities to the final results.

1. INTRODUCTION

In recent years, automatic assistance and safety systems for supporting elderly people at their homes have gained increasing research interest [1, 2]. Multi-modal video cameras, microphones and computer processing power have become powerful and cheap enough to potentially allow for full time surveillance and assessment of the home environment.

In this paper we present a system which can automatically track multiple people and simultaneously detect unusual events. For this purpose, a smart home environment, equipped with multiple multi-modal sensors is used. Our method is twofold: First (Section 2), data from all available sensors is fused in a 3D voxel occupancy grid, where we apply graph cuts [3] to accurately reconstruct the 3D scene. Our algorithm is capable of fusing information from CCTV, thermal, infrared, and PMD-range cameras.

Secondly (Section 3), we apply a Viterbi tracking algorithm, which not only tracks every person, but simultaneously detects whether they are standing, or have fallen down. Depending on the tracking output, the system is capable of detecting events such as *entering*, *exiting*, *sitting on the sofa* and most importantly falling to the floor. We evaluate different configurations and show excellent recognition rates on the PROMETHEUS Smart Home database [4] in Section 4. Section 5 concludes the paper.

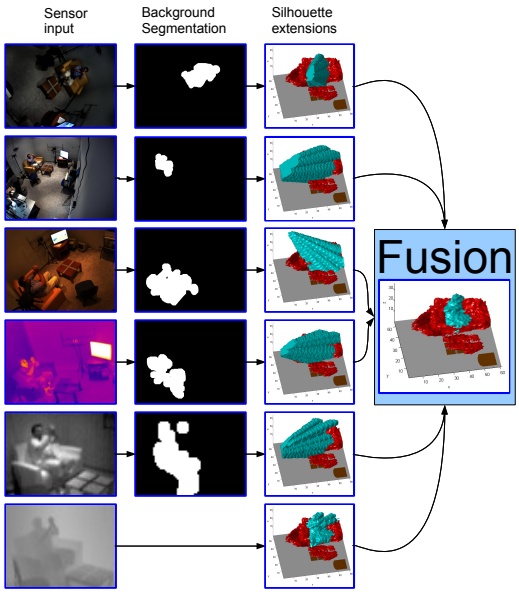

Fig. 1. The 3D Voxel occupancy grid is calculated as follows: For each input sensor, foreground silhouettes are generated using background subtraction. The visual hulls of the foreground silhouettes are fused in the 3D space using graph cuts. The range channel is a special case: Instead of using background subtraction, the visual hull is generated using the range information.

2. 3D VOXEL OCCUPANCY GRID

In the following, we present an approach for reconstructing the 3-dimensional shapes of objects from multiple multi-modal distributed camera sources. To this end, the scene is quantized to a three dimensional voxel occupancy grid. The occupancy of each of the voxels is determined by the joint observations of all available sensors. The method is able to utilize heterogeneous sensors such as CCTV, thermal, infrared and range sensors.

2.1. Definition and Sensor Projection

The reconstruction of the scene is done in a 3-dimensional occupancy grid V, with a neighborhood system $N \subset V \times V$ which connects each voxel to its adjacent voxels. For every voxel $v \in V$ there is a binary labeling f_v which is 1, if the voxel is occupied, and 0, if it is not occupied.

The input to the algorithm comes from multi-modal visual sensors, i.e. visual cameras, infrared cameras as well as the photonic mixer device (PMD) which produces a near infrared image (NIR) and a range image. We distinguish between two principal categories of sensors: intensity and range. There are m intensity sensors and n range sensors. In our experiments we have $m = 4$ intensity sensors and $n = 1$ range sensors.

The set of pixels in camera k is denoted as C_k. We use background modeling [5] to determine, which of the pixels in C_k are foreground. The subset $F_k \subset C_k$ denotes all the pixels which are determined to be foreground by the background modeling method. All of the cameras are calibrated using the Tsai camera calibration method [6]. With the use of this calibration, each pixel $c_k \in C_k$ of camera k intersects with a set of voxels, which is denoted as $V(c_k) \subset V$. Consequently, each voxel v corresponds to a multitude of pixels in the corresponding camera k. The visual observation set $O(v)$ describes for each voxel v, which sensors see it as foreground:

$$O(v) = \left\{k | \exists c_k \text{ with } c_k \in V^{-1}(v) \wedge c_k \in F_k\right\} \qquad (1)$$

The range sensor is a special case. The function $r(v)$ describes, if the voxel v is foreground, based on the range information.

$$r(v) = \begin{cases} 1 & \text{range}(V^{-1}(v)) < \text{dist}(v, \text{PMD}) \\ 0 & \text{else} \end{cases} \qquad (2)$$

Here $\text{dist}(v, \text{PMD})$ denotes the Euclidean distance from the voxel v to the center of the PMD camera and $\text{range}(c_k)$ denotes to distance measured with the PMD device at pixel c_k.

2.2. Fusion using Graph Cuts

Initial experiments have shown, that a simple fusion method using intersection of visuall hulls, does not suffice in many cases. To further improve the reconstruction quality, we used a global energy function with a data term $D_v(f_v)$ and a smoothness term $S(f_v, f_{v'})$. This allows to naturally include a smoothness constraint. Minimizing this energy function is superior to silhouette intersection, which has no means of incorporating a smoothness term. The energy function is given as

$$E(f) = \sum_{v \in V} D_v(f_v) + \mu \sum_{v,v' \in N(v)} S(f_v, f_{v'}) \qquad (3)$$

For each voxel v, the data term $D_v(f_v)$ assigns a cost depending on the label f_v. We define the visibility ratio $h(v) = \frac{\|O(v)\| + r(v)}{\|\hat{O}(v)\|}$, $\hat{O}(v) = \left\{k | \exists c_k \text{ with } c_k \in V^{-1}(v)\right\}$, which defines for each voxel the ratio of the number of cameras observing the voxel as foreground divided by the total number of cameras which can see the voxel. This is an important measure, because voxels can be oberserved by a variable number of cameras. We then define the data term as follows:

$$D_v(f_v) = \begin{cases} h(v) & f_v = 1 \\ 1 - h(v) & f_v = 0 \end{cases} \qquad (4)$$

The smoothness term is defined on the close neighborhood as:

$$S(f_v, f_{v'}) = \begin{cases} 0 & f_v = f_{v'} \\ 1 & \text{else} \end{cases} \qquad (5)$$

The final labeling $f = \arg\min_f E(f)$ is obtained using graph cuts [7]. In all our experiments we set $\mu = \frac{1}{100}$ in Equation 3. This factor weighs the influence of the data term versus the smoothness term.

3. EVENT TRACKER

The basic idea of the event tracker is to formulate the event detection stage jointly with the tracking stage. In other words, we use a multi object tracking algorithm, which not only tracks all the people in the scene, but simultaneously tracks the configuration (standing or fallen down) of the person.

3.1. Viterbi Formulation

We use a maximum a posteriori method, more specifically the Viterbi algorithm, to find the optimal trajectories. To begin, we will first introduce the state variables. At each time t, the state of a person i is given by $S_t^i = \left\{x_t^i, y_t^i, l_t^i\right\}$. Thus the state not only contains the position (x_t^i, y_t^i), but an additional flag l_t^i which can hold one of three values: $l_t^i \in \{outside, standing, fallen\}$. We denote the joint state space of all N people at time t by $\mathbf{S}_t = \left\{S_t^1, \ldots, S_t^N\right\}$. $\mathbf{S}^i = \left\{S_1^i, \ldots, S_T^i\right\}$ denotes the trajectory of person i. The complete state of the full sequence containing T frames is then given by $\mathbf{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_T)$.

Given the set of observations $\mathbf{I} = (I_1, \ldots, I_T)$, we seek to maximize the state sequence $\mathbf{S}$, given the observations $\mathbf{I}$.

$$\hat{\mathbf{S}} = \arg\max_S P(\mathbf{S}|\mathbf{I}) \qquad (6)$$

Because we already have a heavily discretized occupancy grid (and thus a rather low number of states), we propose to use the Viterbi algorithm to find the optimal state sequence. The Viterbi algorithm is an iterative algorithm, which at each time step returns the optimal trajectory up until this time step. However, despite the discretized occupancy grid, the optimal solution is intractable, because the number of states increases exponentially with a higher number of people.

A solution to this problem is to compute the trajectories for each person one after the other:

$$\hat{\mathbf{S}}^1 = \arg\max_{\mathbf{S}^1} P(\mathbf{S}^1|\mathbf{I}) \quad (7)$$

$$\hat{\mathbf{S}}^2 = \arg\max_{\mathbf{S}^2} P(\mathbf{S}^2|\mathbf{I}, \hat{\mathbf{S}}^1) \quad (8)$$

$$\vdots$$

$$\hat{\mathbf{S}}^N = \arg\max_{\mathbf{S}^N} P(\mathbf{S}^N|\mathbf{I}, \hat{\mathbf{S}}^1, \hat{\mathbf{S}}^2, \ldots, \hat{\mathbf{S}}^{N-1}) \quad (9)$$

This means that the optimization of a trajectory is conditioned on the results from optimizing all the previous trajectories. The conditioning implies that trajectories cannot use locations which are already occupied by other trajectories.

Optimizing a single trajectory then becomes a matter of running the standard Viterbi algorithm [8]. We need to find the most likely path through the state sequence, which maximizes the posterior probability of Equation 6. This is achieved with an iterative procedure. At each time t,

$$\Psi_t(k) = \max_{S_1,\ldots,S_{t-1}} P(I_1, \ldots, I_t, S_1, \ldots, S_{t-1}, S_t = k) \quad (10)$$

denotes the maximum probability of ending up in state k at time t. With the Markov assumptions, the current state is only dependent on the previous state $P(S_t|S_{t-1}, S_{t-2}, \ldots) = P(S_t|S_{t-1})$ and the observations are independent given the state $P(\mathbf{I}|\mathbf{S}) = \prod_t P(I_t|S_t)$. Therefore the iterative Viterbi equation can be written as:

$$\Psi_t(k) = P(I_t|S_t = k) \max_{\lambda} P(S_t = k|S_{t-1} = \lambda)\Psi_{t-1}(\lambda) \quad (11)$$

The maximization operator in Equation 11 finds the optimal predecessor in frame $t-1$ when going to state k at time t. Thus a backtracking starting from the final optimum at $t = T$ yields the optimal trajectory.

3.2. Motion Model

The motion model is given by the term $P(S_t = k|S_{t-1} = \lambda)$. It is the probability of entering state k, if the system was in state λ in the previous time step. We model this probability with a Gaussian distribution centred at λ and with a standard deviation of $\sigma = 100mm$. This way, motions of approximately 0.1 meter per time step and less are encouraged, while bigger motions are not likely (but still possible).

3.3. Appearance Model

Here $P(I_t|S_t = \lambda)$ is the observation model. Given a hypothesized state $S_t = \lambda$, this is the likelihood of observing that state. This observation likelihood is determined as follows: First we correlate the 3D voxel occupancy grid with two templates. One template is a thin and tall cylinder, representing a standing human (see Figure 2(top row)). The other template is a flat and wide template representing a fallen human (see Figure 2(bottom row)). Then the output from the correlation is normalized to a probability distribution.

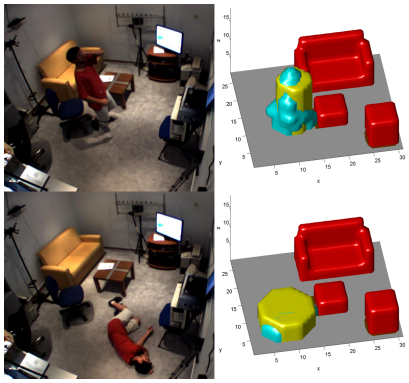

Fig. 2. Falling person and the corresponding 3D voxel occupancy grid. Background objects (red); Observations (blue); Human correlation templates at the tracked position (yellow)

3.4. Event Detection

Detecting events becomes very straight forward after the informative output from the tracking module. The tracker gives for each person the position and the standing/fallen flag. Setting appropriate thresholds on the position readily gives results for *entering*, *exiting*, *sitting down* and *standing-up*. The event *falling down* can be directly detected from the standing/fallen flag l_t.

4. EXPERIMENTS

To evaluate our method, we use the PROMETHEUS indoor smart home database [4]. This multi-sensors, multi-modal database shows daily scenarios in a living room.

4.1. Evaluation of Tracking

In order to evaluate different configurations of sensors, we set up 7 experiments (Figure 1), each of which only uses a subset of the five available sensors.

The tracking performance is assessed using four measures: Multiple Object Tracking Mean Error (MOTME), Multiple Object Tracking Variance (MOTV), Missed Tracks per Frame (MTPF), and Artifact Tracks per Frame (ATPF).

In order to computer MOTME and MOTV, we first compute a distance matrix at each frame between all found objects and all ground truth objects. The Hungarian algorithm is used to optimally match found and ground truth objects. However, when the match distance exceeds 1 meter, we stop matching. Once matched, mean and variance are easily computed. Undetected persons and erroneously detected persons are mea-

	Cam1	Cam2	Cam3	Thermal	PMD
Experiment 1	x	x	x	x	x
Experiment 2	x	x	x	x	
Experiment 3	x	x	x		x
Experiment 4	x	x	x		
Experiment 5	x	x			x
Experiment 6		x	x		x
Experiment 7	x		x		x

Table 1. Performance results for the five detectable events

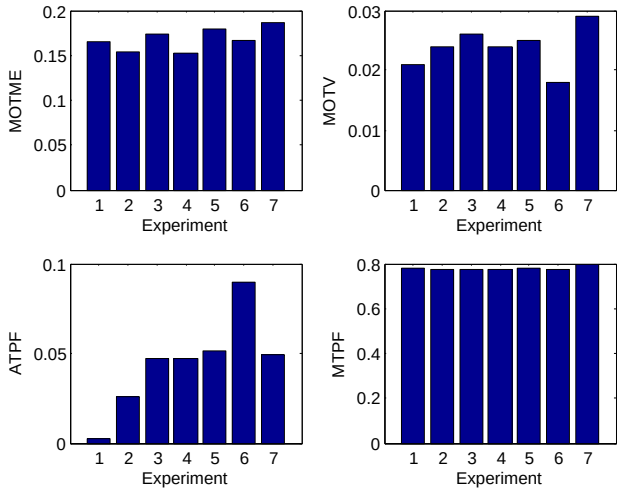

Fig. 3. Tracking results

sured by the missed tracks per frame and artifact tracks per frame measures, respectively.

Tracking results are shown in Figure 3. Regarding MOTME, MOTV and MTPF, all configurations seem to perform roughly similarly with only minor differences. Most notably the ATPF rate is dramatically reduced when using all available sensors. It can furthermore also be seen, that in terms of Mean Error, the configurations without the PMD sensor slightly outperform the others. Also experiment 6 sticks out: While the tracking variance is low, it suffers at the same time from a high artifact rate. Thus, camera 1 seems to have a good view of the scene and is important for good results.

4.2. Evaluation of Event Detection

For evaluation, the processing results are compared against manually annotated ground truth. In order to account for annotation errors and detection uncertainty, we allow a temporal window of $\Delta f = 30$ frames ($\widehat{=}$ 2sec @15fps) for matching ground truth to detection results.

The final results of our event detection method are shown in Table 2. It can be seen that the event *falling down* has been recognized with 100% recall and a few false positives. A few false positives are admissible because in safety applications, the focus is on a high recall rate. In our experiments a few false positives occured when people leaned down to help up a person who has fallen down before. Our algorithm is able

	Precision	Recall	F1-measure
falling down	85.7%	100%	96.1%
sitting down	100%	100%	100%
standing up	100%	100%	100%
entering home	83.2%	71.4%	76.8%
exiting home	80.0%	66.6%	72.7%

Table 2. Performance results for the five detectable events

to detect the *sitting down* and *standing up* events with perfect precision and recall. The *entering* and *exiting* events are harder to detect, especially because in our dataset, people often enter or exit the scene in groups of two or three.

5. CONCLUSION

In this paper we have shown how data from multiple, heterogeneous image sensors can be efficiently combined to detect a number of events with application to surveillance in a smart home environment. Furthermore, we demonstrated simultaneous tracking and event detection using an extended multi-object Viterbi tracking framework. On the Prometheus Smart Home database we showed the impact of multiple sensor configurations on the tracking performance and we showed excellent event detection results.

6. REFERENCES

[1] Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee, "Robust fall detection using human shape and multi-class support vector machine," in *Proc. Indian Conf. on Computer Vision, Graphics & Image Processing*, 2008, pp. 413–420.

[2] M. Shoaib, T. Elbrandt, R. Dragon, and J. Ostermann, "Altcare: Safe living for elderly people," in *4th Int. ICST Conf. on Pervasive Computing Technologies for Healthcare*, 2010.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pat. Analysis and Machine Intelligence*, vol. 23, no. 11, 2001.

[4] Stavros Ntalampiras, Dejan Arsić, Andre Störmer, Todor Ganchev, Ilyas Potamitis, and Nikos Fakotakis, "PROMETHEUS database: A multi-modal corpus for research on modeling and interpreting human behavior," in *Proc. Int. Conf. on Digital Signal Processing*, 2009.

[5] Zoran Zivkovic and Ferdinand van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recogn. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[6] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3-D machine vision," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1986, pp. 364–374.

[7] Vladimir Kolmogorov and Ramin Zabih, "What energy functions can be minimized via graph cuts?," in *Proc. European Conf. on Computer Vision*, 2002, pp. 65–81.

[8] G. D. Fornay, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.