EVENT DETECTION IN A SMART HOME ENVIRONMENT USING VITERBI FILTERING AND GRAPH CUTS IN A 3D VOXEL OCCUPANCY GRID

Martin Hofmann, Moritz Kaiser, Nico Lehment, Gerhard Rigoll
Institute for Human-Machine Communication, Technische Universität München, Arcisstr. 21, Munich, Germany
$\{$*martin.hofmann, kaiser, lehment, rigoll*$\}$*@tum.de*

Keywords: event detection, smart home, voxel occupancy grid, graph cuts, viterbi tracking

Abstract: In this paper we present a system for detecting unusual events in smart home environments. A primary application of this is to prolong independent living for elderly people at their homes. We show how to effectively combine information from multiple heterogeneous sensors which are typically present in a smart home scenario. Data fusion is done in a 3D voxel occupancy grid. Graph Cuts are used to accurately reconstruct people in the scene. Additionally we present a joint multi object Viterbi tracking framework, which allows tracking of all people, and simultaneously detecting critical events such as fallen persons.

1 Introduction

In recent years, automatic assistance and safety systems for supporting elderly people at their homes have gained increasing research interest. 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 order to prolong independent living at home, specialized communication equipment has long been utilized to allow a person to call for assistance when needed. However, typically a person in need of help is unable to call for help or press the alarm button himself. Thus it is desirable to have a surveillance system, which automatically detects such a crucial condition and calls for assistance in case of an accident.

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

Secondly, 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 and based on meta data, the system is capable of detecting events such as ”entering”, ”exiting”, ”sitting on the sofa” and most importantly falling to the floor.

We begin with a review of related work. In Section 3, we present the 3D voxel occupancy grid and the reconstruction using Graph Cuts. Section 4 presents our extension to the Viterbi Tracking algorithm for event detection. Experiment on the PROMETHEUS dataset (Ntalampiras et al., 2009) are shown in Section 5 and we conclude in Section 6.

2 Related Work

The systems currently deployed to elderly people are non intelligent and require manual interaction. The person has means of sending an emergency call, such as a remote transponder, however in the event of an accident the person has to be conscious and must be able to press an emergency button.

There exists quite a number of automated systems which use environmental sensors. For example (Foroughi et al., 2008) uses Support Vector Machine classification on human silhouette shape changes. However their system requires a very specific camera setup in front of a monochrome background, which is unnatural for real world scenarios. In (Diraco et al., 2010) a single Photonic Mixer Device is used to find the centroid of a person. Similarly in (Shoaib et al., 2010) the authors classify on the foreground blobs extracted from a single camera with monochrome background.

Thus most of the current approaches use data from a limited number of visual sensors or need very specific camera setup. Our approach in contrast is capable of fusing data from a multitude of distributed and heterogeneous sensors into one common scene reconstruction. Tracking and event detection in this joint observation space is much more robust and leads to better results because the reasoning can be done in 3D space.

For 3D reconstruction we decided to use a 3D voxel occupancy grid. 3D voxel occupancy dates back the early 1980's (Martin and Aggarwal, 1983). This method uses silhouettes from multiple views and projects them back into the voxel grid where the shapes of the objects are inferred by intersection of the back projected silhouettes. The method is therefore often also called shape-from-silhouette (Cheung et al., 2003)(Cipolla and Blake, 1992).

However, the general shape-from-silhouette method is very sensitive to noise and errors in the generation of the silhouettes. For example, if a pixel in the foreground silhouette from one view is falsely detected as background, the resulting 3D structure will typically have a hole. To avoid this, morphological operations (Serra, 1983) are often applied to the silhouettes or the 3D volume to enhance the quality. However, morphological operations typically introduce deforming artifacts.

In order to avoid artifacts and to get a more precise reconstruction, we use a soft fusion of the silhouettes similar to (Snow et al., 2000). To this end, a global energy function is defined on the 3D voxel occupancy grid. This energy function contains a data term, which alone is equivalent to the standard shape-from-silhouette method. Additionally, the energy contains a smoothness term, which in essence allows to pad holes without introducing artifacts.

As mentioned in the introduction, we use an extended state tracker to follow people and to find out if they are standing, sitting or fallen down. We use an extension to the Viterbi tracking algorithm. This is similar to (Fleuret et al., 2007). However, the main difference is that we extend the state space to not only include the (x_t, y_t) coordinates, but also a flag l_t, which states whether the person is standing, or has fallen down.

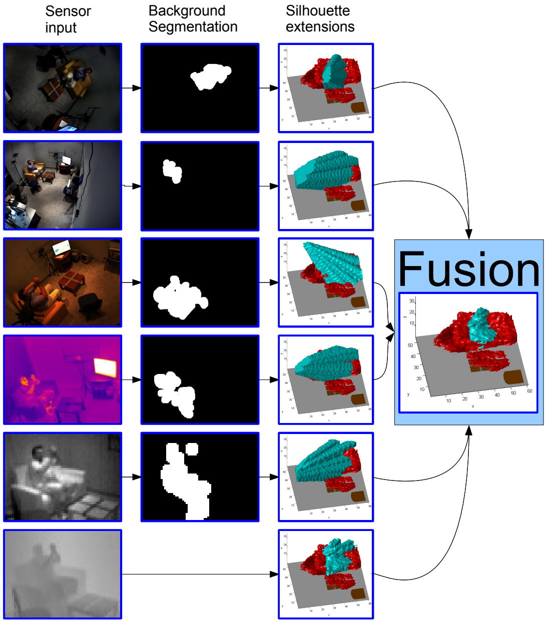

Figure 1: The 3D voxel occupancy grid is calculated as follows: For each input sensor, foreground silhouettes are generated using background subtraction. The visual hulls of the foreground silhouettes are fused in the 3D space using either intersection of the visual hulls or graph cuts. The range channel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

3 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.

With this method it is possible to generate accurate 3D detections, as well as an accurate reconstruction of the scene. All the processing is done in 3D and therefore allows reasoning in a 3D environment as opposed to just flat 2D camera images. The proposed method is especially applicable to short range environments just like the smart home indoor scenarios.

We show two methods for fusing data: One is by intersection of the visual hulls. The other uses graph cuts (Boykov et al., 2001). The later gives much better results.

3.1 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 multimodal 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 $m = 5$ intensity sensors and $n = 1$ range sensors. Details of our specific setup can be found in Section 5.1.

The set of pixels in camera k is denoted as C_k. We use background modeling (Zivkovic and van der Heijden, 2006) 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 (Tsai, 1986). 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:

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

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

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

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

3.2 Fusion using Intersection

First experiments of fusing projected data in the 3D voxel occupancy grid is by intersection of the visual hulls. The fusion then efficiently becomes:

$$f_v = \begin{cases} 1 & \|O(v)\| + r(v) \geq k + 1 - \mu \\ 0 & \text{else} \end{cases}, \quad \text{with } \mu = 0 \quad (3)$$

This means that a voxel v is labeled as foreground if it can be observed as foreground by all visual sensors, and if in addition the voxel can be explained as foreground by the PMD range channel.

In the case that detections are erroneous and suffer from misses, the proposed algorithm can easily be made more robust by setting $\mu > 0$. In this case only $k - \mu$ out of the available views have to intersect, which leads to better detections at the cost of more false positives.

3.3 Fusion using Graph Cuts

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

$$E(f) = \sum_{v \in V} D_v(f_v) + \mu \sum_{v,v' \in N(v)} S(f_v, f_{v'}) \quad (4)$$

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:

$$D_v(f_v) = \begin{cases} h(v) & f_v = 1 \\ 1 - h(v) & f_v = 0 \end{cases} \quad (5)$$

The smoothness term is defined on the close neighborhood as:

$$S(f_v, f_{v'}) = \begin{cases} 0 & f_v = f_{v'} \\ 1 & \text{else} \end{cases} \quad (6)$$

The final labeling $f = \arg\min_f E(f)$ is obtained using graph cuts (Kolmogorov and Zabih, 2002). In all our experiments we set $\mu = \frac{1}{100}$ in Equation 4. This factor weighs the influence of the data term versus the smoothness term.

4 Event Tracker

In this section we present a combined low and high level action recognition framework for smart home scenarios. The basic idea 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. The output of this extended tracking framework is rich enough to detect a multitude of events without further classification or recognition steps.

The event tracker takes input solely from the observation space generated using the 3D voxel occupancy grid described in the previous section. This 3D reconstruction proved highly accurate and robust for the purpose of event detection.

The main focus in this work is on detecting fallen persons. However, as a byproduct, the method is also capable of recognizing four additional events. More specifically, ”Sitting down”, ”Standing up”, as well as ”Entering home” and ”Exiting home”.

4.1 Viterbi Formulation

In this section we present the optimization method of the joint tracking and event detection framework. 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 $S_t^i = \{x_t^i, y_t^i, l_t^i\}$. 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 by $\mathbf{S}_t = \{S_t^1, \ldots, S_t^N\}$ the joint state space of all people at time t. Here N is the maximum number of people that can potentially be in the scene. Consequently, $\mathbf{S}^i = \{S_1^i, \ldots, S_T^i\}$ 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 $\mathbf{I}$.

$$\hat{\mathbf{S}} = \arg\max_S P(\mathbf{S}|\mathbf{I}) \tag{7}$$

There are many possibilities to optimize such a posterior, for example using particle filters such as the condensation algorithm or Markov Chain Monte Carlo methods. 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:

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

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

$$\vdots$$

$$\hat{\mathbf{S}}^N = \arg\max_{\mathbf{S}^N} P(\mathbf{S}^N|\mathbf{I}, \hat{\mathbf{S}}^1, \hat{\mathbf{S}}^2, \ldots, \hat{\mathbf{S}}^{N-1}) \tag{10}$$

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. Assuming perfect observations this sequential approach does not do any harm and the optimal trajectories can still be obtained. However in practice, the observations suffer from spurious detections, miss detections and noise. Obviously the described greedy optimization fails in these cases.

An elegant solution(Fleuret et al., 2007) is to process the data in temporal batches of a certain length $T = T_0$. The basic idea is as follows: After the optimization (Equations 8-10) is performed, only the first 10% of the tracking results are kept and the rest is discarded. The temporal batch is then shifted forward. The trajectories are sorted such that for the next optimization the trajectory with the highest confidence is computed first. This ensures that trajectories with stable observations are optimized first, while trajectories with weak observations cannot interfere with the stable trajectories.

Optimizing a single trajectory then becomes a matter of running the standard Viterbi algorithm (Fornay, 1973). We need to find the most likely path through the state sequence, which maximizes the posterior probability:

$$\hat{\mathbf{S}} = \arg\max_S P(\mathbf{S}|\mathbf{I}) = \arg\max_S P(I_1, \ldots, I_T, S_1, \ldots, S_T) \tag{11}$$

This is achieved with an iterative procedure. At each time t,

$$\Psi_t(k) = \max_{S_1, \ldots, S_{t-1}} P(I_1, \ldots, I_t, S_1, \ldots, S_{t-1}, S_t = k) \tag{12}$$

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 $P(\mathbf{I}|\mathbf{S}) = \prod_t P(I_t|S_t)$.

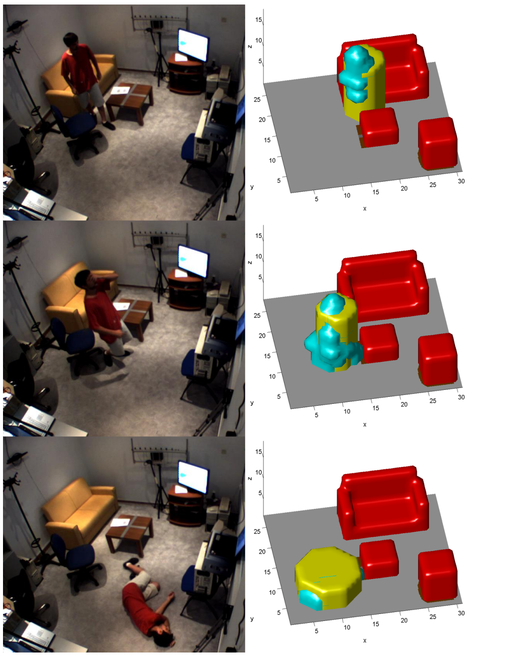

Figure 2: Sequence of a falling person and the corresponding observation in the 3D voxel occupancy grid. The 3D voxel occupancy grid is correlated with a tall thin template, and a flat wide template to get the observation likelihood $P(I|S)$. Manually modeled background objects (red); Observations in the 3D voxel occupancy grid (blue); human correlation templates at the tracked position (yellow).

Therefore the iterative Viterbi equation can be written as:

$$\Psi_t(k) = P(I_t|S_t = k) \max_{\lambda} P(S_t = k|S_{t-1} = \lambda)\Psi_{t-1}(\lambda) \tag{13}$$

The maximization operator in Equation 13 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.

4.2 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).

4.3 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.

4.4 Event Detection

Detecting events becomes very straight forward after the rich 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 give results for ”entering”, ”exiting”, ”sitting down” and ”standing-up”. For example, a trajectory which starts close to the entrance region will invoke the ”entering” event. The event ”falling down” can be directly detected from the standing/fallen flag l_t. The event is detected at each transition from $l_t = standing$ to $l_{t+1} = fallen$.

5 Experiments

5.1 Database and Setup

The data used in this paper has been recorded during the 7th Framework EU Project PROMETHEUS (Ntalampiras et al., 2009). We use data from two smart home indoor scenarios. In these scenarios up to 4 individuals, played by actors, portray daily behavior in a living room. The data is therefore captured in a very controlled environment.

The database contains two recordings which, besides others, contain several instances of the falling down events. The recordings are 15min and 30min respectively. The scenario is recorded from five cameras with a total of six image channels. These include three visual channels from CCTV cameras, one thermal channel from a thermal camera, as well as a near infrared (NIR) channel and a range channel from a Photonic Mixer Device. We divide the six channels into two different kinds of data sources (intensity and range), as described in Section 3.1. The $m = 5$ intensity sensors consist of the three visual cameras, the thermal camera and the NIR channel of the PMD device. The PMD device generates the $n = 1$ range channel.

The visual cameras deliver high resolution (1024×768 and 768×576) color images. The thermal camera has a medium resolution of 240×320. The PMD sensor, a CMOS time of flight Camera

	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 1: Performance results for the five detectable events

(PMD[Vision]3k-S), delivers a range image with a spatial resolution of 64×48 pixels for a range of up to 7.5m in a depth resolution of $\approx$ 1cm as well as a 64×48-pixel NIR (Near-infrared) image.

5.2 Evaluation and Results

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 for matching ground truth to detection results.

The final results of our event detection method are shown in Table 1. 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 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.

6 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. We have shown that for fusing multiple heterogeneous data sources, a 3D voxel occupancy grid is beneficial. Furthermore, we demonstrated simultaneous tracking and event detection using an extended multi-object Viterbi tracking framework. We applied our method to the multi-camera, multi-modal Prometheus smart home database. In this specific application, our algorithm is capable of detecting falling people and a number of other events. We showed excellent results on this smart home database and showed that the proposed application setup can in fact be used for assistance systems for the elderly.

REFERENCES

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pat. Analysis and Machine Intelligence*, 23(11):1222–1239.

Cheung, G. K., Baker, S., Simon, C., and Kanade, T. (2003). Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-from-silhouette with stereo. *Comp. Vis. Pat. Rec.*

Cipolla, R. and Blake, A. (1992). Surface shape from the deformation of apparent contours. *Int. J. Comput. Vision*, 9(2):83–112.

Diraco, G., Leone, A., and Siciliano, P. (2010). An active vision system for fall detection and posture recognition in elderly healthcare. In *Design Automation & Test*, pages 1536–1541.

Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2007). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Trans. Pat. Analysis and Machine Intelligence*.

Fornay, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Foroughi, H., Rezvanian, A., and Paziraee, A. (2008). Robust fall detection using human shape and multi-class support vector machine. In *Proc. Indian Conf. on Computer Vision, Graphics & Image Processing*, pages 413–420.

Kolmogorov, V. and Zabih, R. (2002). What energy functions can be minimized via graph cuts? In *Proc. European Conf. on Computer Vision*, pages 65–81.

Martin, W. and Aggarwal, J. (1983). Volumetric descriptions of objects from multiple views. *IEEE Trans. Pat. Analysis and Machine Intelligence*, 5(2):150–158.

Ntalampiras, S., Arsić, D., Störmer, A., Ganchev, T., Potamitis, I., and Fakotakis, N. (2009). PROMETHEUS database: A multi-modal corpus for research on modeling and interpreting human behavior. In *Proc. Int. Conf. on Digital Signal Processing*.

Serra, J. (1983). *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA.

Shoaib, M., Elbrandt, T., Dragon, R., and Ostermann, J. (2010). Altcare: Safe living for elderly people. In *4th Int. ICST Conf. on Pervasive Computing Technologies for Healthcare 2010*, volume 0.

Snow, D., Viola, P., and Zabih, R. (2000). Exact voxel occupancy with graph cuts. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 345–352.

Tsai, R. (1986). An efficient and accurate camera calibration technique for 3-D machine vision. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 364–374.

Zivkovic, Z. and van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780.