

# Dynamic 3D Scene Analysis for Acquiring Articulated Scene Models

Agnes Swadzba, Niklas Beuter, Sven Wachsmuth, and Franz Kummert

**Abstract**—In this paper we present a new system for a mobile robot to generate an articulated scene model by analyzing complex dynamic 3D scenes. The system extracts essential knowledge about the foreground, like moving persons, and the background, which consists of all visible static scene parts. In contrast to other 3D reconstruction approaches, we suggest to additionally distinguish between static parts, like walls, and movable objects like chairs or doors. The discrimination supports the reconstruction process and additionally, delivers important information about interaction objects. Here, the movable object detection is realized object independent by analyzing changes in the scenery. Furthermore, in the proposed system the background scene is feed-backed to the tracking part yielding a much better tracking and detection result which improves again the 3D reconstruction. We show in our experiments that we are able to provide a sound background model and to extract simultaneously persons and object regions representing chairs, doors, and even smaller movable objects.

## I. INTRODUCTION

Bottom-up learning of spatial awareness is an essential capability for robots in the field of service and household robotics as they have to deal with and communicate about unknown and changing environments. To get information about the surrounding, the robot has to perceive, to analyze, and to segment its environment in meaningful parts. This is an inherent 3D interpretation task. First, it has to detect and track the human as its focused interaction partner. Further, the robot should gather information about the static scene (like walls or tables) that do not change their position, and movable objects (like a chair or a door) that can change their position. Instead of building a complex ontology of indoor rooms that describes which scene parts are static and which may move and equipping the robot with diverse (possibly error-prone) object detectors, we propose a light-weight learning methodology which enables the robot to acquire a so-called *articulated scene model* from observing a dynamic scene, directly in 3D. It arises after a few seconds of observation and models the scene in a basic and general way.

This model consists of three components. First, humans and their movements are tracked by a particle filtering approach using a weak object model. Humans are independent entities with regard to the underlying scene, but they are essential for understanding the functional parts of a scene. Second, the model contains movable objects as articulated parts of the model (like chairs, doors, or soft toys). These

objects are characterized by the fact that they can change their position in the scene caused by an agent but not by themselves. Our approach is able to detect any object that has been moved without utilizing knowledge about the object itself, its trajectory, or an object model. This component is the most challenging one as the robot does not know beforehand which part of the scene will change its position. Additionally, these movable objects are hard to track as the distinction between the person and the object carried by the person is not easily possible as well as the distinction between the object and the static background when placed back in the scene. Third, the static background that did not change during observation is modeled as the static scene.

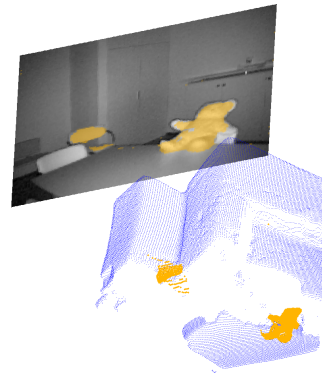


Fig. 1: The orange colored 3D points refer to the acquired articulated scene parts and the blue points to the static background.

The presented approach is based on 3D time-of-flight data which is extended to 6D data with 3D velocity vectors computed using optical flow. Differences between the current frame and the static background are used to determine potential dynamic parts which are refined to moving objects, e.g., persons, in the subsequent tracking step. By excluding these dynamic objects from the current frame, static scene parts are determined which are separated in the subsequent modeling step into static background and static movable objects. These are distinguished by using the assumption that the farthest measurement seen determines the background. Figure 1 shows such a resulting articulated scene model with the blue 3D points holding the static scene and the orange points the movable objects modeled as articulated scene parts.

In contrast to background subtraction methods which integrate or remove objects that appear or disappear over time into their background model we emphasize the necessity to make a distinction between the static background scene and such movable objects. The benefit of our approach is the additional general information about these object regions. There are several advantages connected with this knowledge. First, the modeling of the static background can be improved by excluding these objects from the reconstruction process. As the background model is passed to the tracking part of our system the tracking of humans can be improved significantly due to the reduced search space. Second, the salient object

Applied Informatics, Bielefeld University  
aswadzba@techfak.uni-bielefeld.de  
nbeuter@techfak.uni-bielefeld.de  
swachsmu@techfak.uni-bielefeld.de  
franz@techfak.uni-bielefeld.de

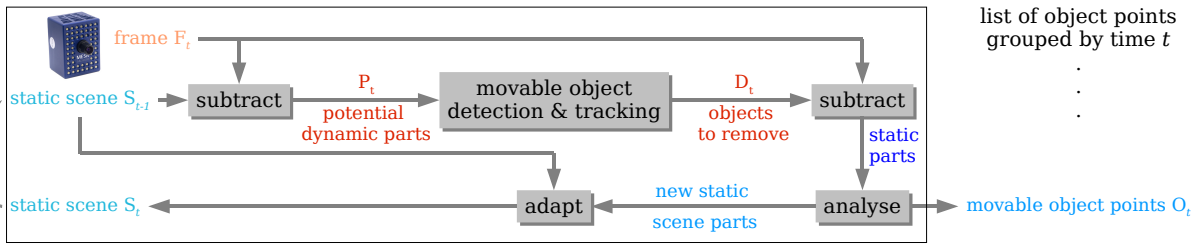


Fig. 2: Flow diagram of our proposed system at time step  $t$ . Each frame is analyzed by comparing the 3D points to the previously known static scene  $S_{t-1}$ . Moving objects are detected and tracked using the potential dynamic parts  $P_t$ . The found objects are subtracted yielding static parts. By analyzing these parts, the movable object points are separated from the actual new static scene parts. The actual static scene  $S_t$  is built by adapting the known static scene  $S_{t-1}$  with the new static scene parts. Finally the static scene  $S_t$  is forwarded to the next frame as previously known static scene.

regions can be utilized to design a pro-active robot. It can link functionalities to them (e.g., the robot should detect the door as an interesting region through which it can leave the room after the door was opened) and learn object characteristics like a grip [1] or a kinematic model [2].

In the following section II, we give a brief overview about previous reconstruction approaches. Afterwards, we explain in III the proposed algorithm, which is detailed in the adjacent sections. In IV, we describe our data acquisition process and the appendant preprocessing methods. The detection and tracking of humans in the scene will be presented more detailed in section V, followed by the concurrent adaptive reconstruction process in VI. Subsequently, we show in VII our experiments which demonstrate the advantage of the simultaneous tracking and reconstruction procedure, and we end with a conclusion about our work in section VIII.

## II. RELATED WORK

As an essential part of dynamic scene analysis, 2D background subtraction has been widely explored in the past years. Based on the GMM formulation of Stauffer and Grimson, Hayman et al. proposed a statistical approach, which can deal with moving foreground [3]. Although the results are promising, they lack in accuracy, because of missing texture, changing illumination, and similar color of fore- and background. In contrast, perceiving the environment in 3D can avoid these constraints and facilitates the analysis of a dynamic scene, especially for a mobile robot, as the depth information yields additional important details. The methods mainly used to acquire 3D information can be divided in passive and active methods. Typical representatives of the passive method are stereo vision systems that usually rely on the principle of establishing correspondences. There exists a wide range of stereo algorithms ranging from classical ones to ones generating dense path maps using Dynamic Programming [4], [5], [6]. Such stereo vision systems can be extended to enhance the point cloud with velocities by individually tracking recognizable 3D points in a six-dimensional position-velocity space [7]. However, stereo vision as a passive system depends on the environmental conditions, this means the appearance of the scene strongly influences the quality of the point cloud generation. Active sensors overcome this restriction by generating and sending a signal on their own and measuring the reflected signal. Laser range scanners deliver one scanning line of accurate

distance measurements often used for navigation tasks [8], [9]. 3D Time-of-Flight (ToF) Sensors [10], [11] combine the advantage of active sensors and camera based approaches as they provide a 2D intensity image and exact distance values in real-time. Compared to stereo rigs the 3D ToF sensors can deal much better with prominent parts of rooms like walls, floors, and ceilings even if they are not textured. In addition to the 3D point cloud, contour and flow detection in the image plane yields motion information that can be used, e.g., for person tracking [12], [13].

Moving objects cause integration errors during the reconstruction of a consistent representation of the static scene background. Detecting moving pixels and excluding them is a first step and shows good results for creating image mosaics [14]. Other approaches extract moving 3D point clouds. Based on such 3D data, localization and tracking of objects can be performed by mean shift clustering of the point cloud [15]. For mobile robots, mapping the environment, and localizing and tracking moving objects can be combined into a single framework [8], [9].

There also are several approaches that extend geometric maps with semantic information. Nüchter et al. [16] introduce a heuristic for extracting scene features like walls, ceilings, and floors. Dedicated objects are detected by trained classifiers. Hois et al. connect the object recognition with logical reasoning using a domain ontology and additional relational scene information [17]. Vasudevan et al. [18] suggest a hierarchical probabilistic representation of space that is based on objects. A global topological representation of places is proposed with object graphs serving as local maps. In contrast to that work, our approach simultaneously tracks dynamic objects, computes the static background, and detects movable object regions independent of pre-known object classes or object specific detection routines.

## III. SYSTEM OVERVIEW

In the given scenario a robot is going to observe its environment with a Swissranger SR3100 which is a 3D Time-of-Flight (ToF) near-infrared sensor delivering in real-time a dense depth map of  $176 \times 144$  pixels resolution [10]. The robot should acquire knowledge about the static background, movable objects, and dynamically moving objects/persons. In this paper the robot is told to observe its environment passively which means the robot camera stays static for a few seconds acquiring in the meantime round about 80

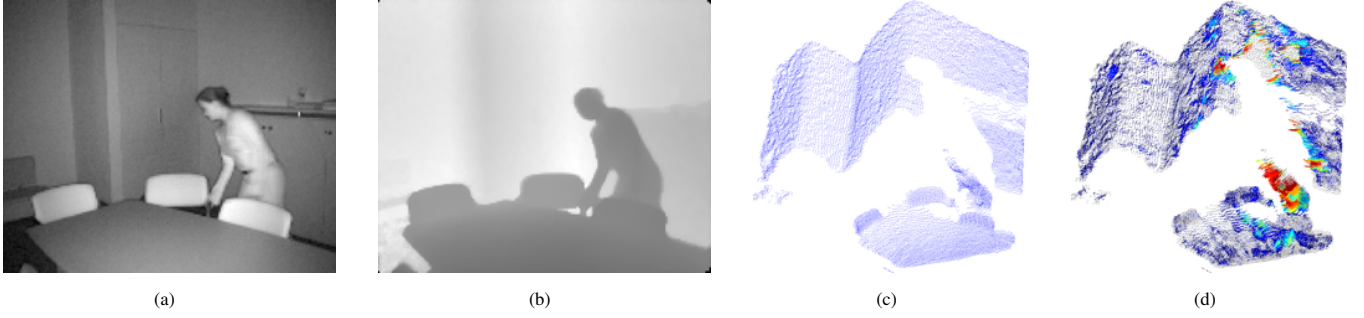


Fig. 3: (a) shows an amplitude image of the Swissranger camera, (b) shows the corresponding depth image, (c) the smoothed 3D point cloud  $F = \{f^i\}$  where invalid measurements were removed, and (d) shows the 3D point cloud annotated with velocity vectors  $V = \{v^i\}$  computed using the optical flow in 2D.

frames. The characteristics of the Swissranger are also used to represent the static scene. The static scene is organized in the same way like a frame of the Swissranger,  $n(= 176 \times 144)$  distances of the background are provided and can be recomputed to  $n$  3D points using the intrinsic parameters of the camera.

An overview of the workflow in our system is given in Figure 2. At each time step  $t$  a detection and tracking of dynamic objects is performed followed by an extraction of static scene parts in the current frame. The tracking is supported by the knowledge about the current static scene generated out of all previous frames. Therefore, in a first step, those points in the current frame

$$F_t = \{\vec{f}_t^i\}_{i=1\dots n} \quad (1)$$

are subtracted which are part of the current static scene

$$S_{t-1} = \{\vec{s}_{t-1}^i\}_{i=1\dots n}. \quad (2)$$

The remaining unknown points

$$P_t = F_t - S_{t-1} \quad (3)$$

are passed to the object detection and tracking part where moving objects are tracked using a simple cylinder object model and particle filtering. These dynamic object points

$$D_t \subset P_t \quad (4)$$

are subtracted in the scene reconstruction part to determine the static points of frame  $F_t$ . Assuming that the farthest measurement per pixel determines the current background, these static parts are compared with the current static scene  $S_{t-1}$  to identify which of these static measurements contribute to the background, which of them define a new static background, and which of them are part of movable objects (once moved but are static at the moment, e.g., chairs, doors, or soft toys). The result is a set of movable object points

$$O_t \quad (5)$$

and a new static scene

$$S_t = \{\vec{s}_t^i\}_{i=1\dots n} \quad (6)$$

for time step  $t$ . The exact calculation of the defined scene parts is described in the subsequent chapters.

#### IV. PREPROCESSING AND MOTION COMPUTING

The data acquired with the Swissranger camera are affected by noise arising from different reflection properties, additional infra-red light in the scene, and measurement errors at edges (so-called “flying pixels”). The distance image (Fig. 3(b)) is smoothed with a distance-adaptive median filter where on each pixel a different mask size (e.g.,  $3 \times 3$ ,  $5 \times 5$ , or  $7 \times 7$ ) is applied depending on the distance value of the pixel. For choosing the right mask, the distance measurement range of the camera (0 to 7.5m) is divided into three equal intervals. The amplitude values (Fig. 3(a)) encode the amount of infra-red light reflected at the corresponding world point. Small values often arise in the case of badly reflecting surfaces. Such unreliable measurements are removed by amplitude thresholding ( $\theta_{\text{amp}} = \sum_{i=1}^n \text{amp}_i$  with  $\text{amp}_i$  representing the amplitude values of the current frame). At last, the “flying pixels” are treated by computing edges (e.g., using a canny edge filter) on the distance image and removing the points located on the edges. The resulting 3D point cloud is presented in Fig. 3(c).

Both steps of our system, the object tracking as well as the scene reconstruction, need a velocity annotated 3D point cloud. The velocity  $\vec{v}^i$  at each point  $f^i$  is used to generate hypotheses about objects and static parts. Due to the fact that for each 3D point cloud a corresponding 2D amplitude image exists the problem of computing 3D velocities can be reduced to the problem of computing 2D velocities. The velocity in depth can be directly calculated using the depth information available for each 2D amplitude pixel. A widely used technique to get 2D velocities from a pair of grayscale images is the optical flow approach introduced by Lucas and Kanade [19]. For each pixel of image  $\mathcal{I}_1$  a corresponding pixel in image  $\mathcal{I}_2$  needs to be computed. Good matches are computed via a type of Newton-Raphson iteration using the spatial intensity gradient. It is assumed that the optical flow is constant within a certain neighborhood  $\mathcal{N}$  which allows to solve the Optical Flow Constraint via least square minimization. Here, we have used a hierarchical implementation of Lucas’s and Kanade’s algorithm written by Sohaib Khan <sup>1 2</sup>. The 3D point cloud with 3D velocity vectors is shown in Figure 3(d).

<sup>1</sup><http://www.cs.ucf.edu/~khan/>

<sup>2</sup><http://server.cs.ucf.edu/~vision/source.html>

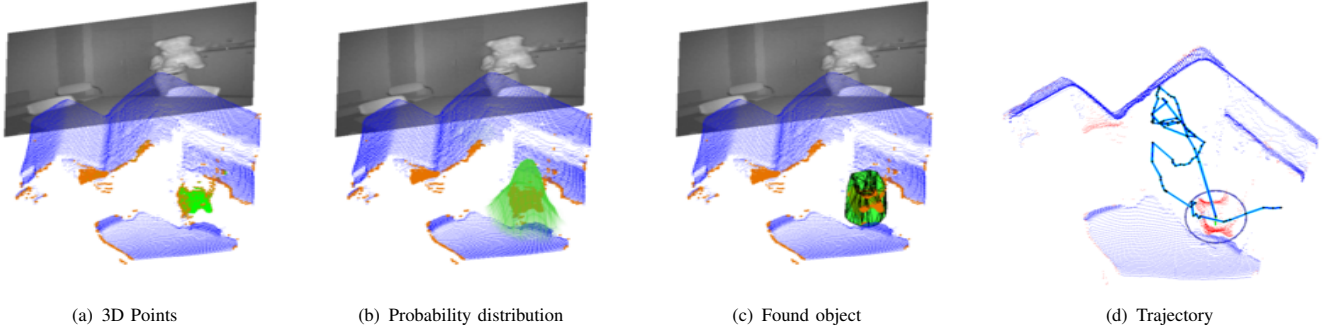


Fig. 4: The images explain the tracking for frame 96 of  $S_{s2,r1}$ . (a) shows the blue points belonging to the static scene  $S_{t-1}$ . The dynamic pixels  $P_t$  are plotted in orange with a green velocity vector. (b) The objects are detected and tracked using the observation function (see Eq. 7). The probability of the particle distribution is plotted in green. (c) The maximum of the observation function denotes the found object (shown as green box). (d) In the birds-eye view of the scene the resulting object trajectory is plotted. The blue circle contains the object at the actual position.

## V. DETECTION AND TRACKING OF DYNAMIC OBJECTS

As we aim at excluding moving objects from the calculation of the scene reconstruction, the next steps are detecting and tracking these objects. Neglecting a tracked object on the whole rather than moving points only is meant to generate better reconstruction results, as we observed that parts of moving objects are not necessarily moving.

To detect and track dynamic objects in the scene several steps have to be accomplished. First of all, the scene representation has to be simplified to reduce the computation time. The potential dynamic points  $P_t$  are clustered using spatial proximity and homogeneity of the velocities. By incorporating velocity information into clustering, we expect an improvement in segmentation of on the one hand moving objects and the static background and on the other hand of several neighboring moving objects without needing strong models. To build the clusters, we apply a hierarchical clustering using the complete linkage algorithm [20], also called furthest neighbor, to describe the distance between two clusters. The clustering procedure deliberately over-segments the scene, generating many small motion-attributed clusters. For each emerging cluster, the following attributes are extracted based on all associated 6D points: The 2D position of the centroid projected on the ground plane, a weight factor based on the number of points, and the mean velocity of all points.

We use a simple cylindric object model with variable radius to group clusters of similar velocities. This weak model offers an object hypothesis  $o(\vec{a})$ , which is suitable for persons and most encountered objects. It is represented by a five-dimensional parameter vector  $\vec{a} = [x \ y \ v_\theta \ v_r \ r]^T$  with  $x$  and  $y$  being the center position of the cylinder with radius  $r$  on the ground plane,  $v_\theta$  denoting the magnitude, and  $v_r$  indicating the direction of the velocity of the object.

We start with generating a set of object hypotheses by partitioning the observed scene with cylinders for initialization and error recovery and by including the tracking results from the previous frame. This set of resulting hypotheses is predicted into the next frame and tracked through a kernel based particle filter [21] as follows.

Based on the position, size and velocity of each object  $o^{t-1}(\vec{a})$  in the last frame  $F_{t-1}$ , the parameters are predicted

for the current frame  $F_t$  utilizing a first order motion model

$$\vec{a}^* = \Phi(\vec{a}, \dot{\vec{a}})$$

creating a new set of hypotheses:

$$o_k^t(\vec{a}^*) \leftarrow \Phi o_k^{t-1}(\vec{a}) \quad , k = 1, \dots, K$$

Each of these  $K$  hypotheses can be seen as a specific point in the parameter space, also called particle. To find the best matching particle to the actual frame, each particle is rated based on the value in the pdf (probability density function)  $\rho$  (Eq. 7), which bases upon the relative position, relative velocity, and weight of all clusters  $l$  within each cylinder  $o_k$  using Gaussian kernels.

$$\rho(o_k) = K_r(o_k) \sum_{l \in o_k} K_d(l, o_k) K_v(l, o_k) \quad (7)$$

The Kernel  $K_r$  keeps the radius in a realistic range, masking out all hypotheses with a too small or too big radius (Eq. 8).

$$K_r(o_k) = \exp\left(-\frac{r(o_k)^2}{2H_{r,\min}^2}\right) - \exp\left(-\frac{r(o_k)^2}{2H_{r,\max}^2}\right) \quad (8)$$

The kernel widths  $H$  are determined empirically. The functions  $r(\cdot)$ ,  $d(\cdot)$ , and  $v(\cdot)$  extract the radius, the 2D position on the ground plane and the velocity of a cluster  $l$  or a hypothesis  $o_k$ . The kernel  $K_d$  reduces the importance of clusters  $l$  further away from the cylinder center (Eq. 9).

$$K_d(l, o_k) = \exp\left(-\frac{\|d(l) - d(o_k)\|^2}{2H_d^2}\right) \quad (9)$$

$K_v$  is masking out clusters having differing velocities (Eq. 10).

$$K_v(l, o_k) = \exp\left(-\frac{\|v(l) - v(o_k)\|^2}{2H_v^2}\right) \quad (10)$$

Eq. 7 is also called the observation function  $\rho(o_k)$  of the particle filter. The outcome is a density approximation based on the object hypothesis and the attributes of the appendant clusters (see Fig. 4(b)). Several mean shift iterations refine the particles to concentrate them at the local maxima of the distribution, which correspond to the actual objects (see Fig. 4(c)).



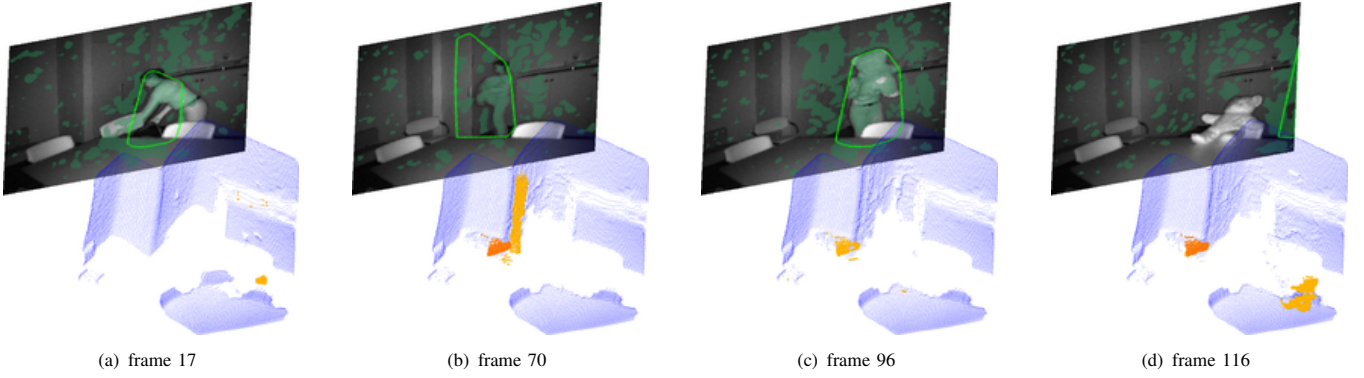


Fig. 5: Here, four example frames of our test sequence  $S_{s2,r1}$  are shown. In this sequence a person takes a chair and places it in the corner, opens a cupboard door, fetches a teddy bear and puts it on the table. For each time step  $t$  the blue colored points are the generated background model  $S_t$ . The orange colored points mark detected movable objects which are actually static but do not belong to the background. The different notes of orange separate points depending on the point in time they have appeared. In the amplitude image of frame  $F_t$  the hulls of objects to remove are drawn in light green and the points annotated with a velocity bigger than a threshold  $\theta_v$  are marked in dark green.

```

1: // Input:
2: // -  $F_t = \{\vec{f}_t^i\}$  (current frame)
3: // -  $S_{t-1} = \{\vec{s}_{t-1}^i\}$  (current background)
4: // -  $D_t$  (current dynamic clusters)
5: // Output:
6: // -  $S_t = \{\vec{s}_t^i\}$  (new background)
7: // -  $O_t$  (movable objects)
8:
9: for  $i = 1$  to  $n$  do
10:   if  $\vec{f}_t^i \notin D_t \wedge |\vec{v}_t^i| < \theta_v$  then
11:     if  $|\vec{s}_{t-1}^i - \vec{f}_t^i| < \theta_d$  then
12:        $\vec{s}_t^i = \vec{s}_{t-1}^i + \frac{1}{w}(\vec{f}_t^i - \vec{s}_{t-1}^i)$ ;
13:       // w: # accumulated values
14:     else
15:       if  $|\vec{f}_t^i| > |\vec{s}_{t-1}^i|$  then
16:          $\vec{s}_t^i = \vec{f}_t^i$ ;
17:       else
18:          $\vec{s}_t^i = \vec{s}_{t-1}^i$ ;
19:          $O_t = O_t \cup \vec{f}_t^i$ ;
20:       end if
21:     end if
22:   end if
23: end for

```

Fig. 6: Algorithm per time step  $t$  for background adaptation and movable object detection.

All 3D points that are associated with the object hypotheses found are marked as dynamic points  $D_t \subset F_t$ . These are passed to the adaptive background modeling process. By assigning an ID to the tracked object, a trajectory can be created to analyze the movement of the object (see Fig. 4(d)).

## VI. ADAPTIVE BACKGROUND MODELLING

This section describes our algorithm for generating the articulated scene model from a complex dynamic scene. Movable objects that form the articulated scene parts are detected and the static background is adapted, simultaneously. Our approach is based on the physical rule that for each pixel the 3D background point is determined by the farthest distance measurement. Due to noise, it is necessary to introduce a threshold  $\theta_d$  above which a change in the distance is significant and does not arise from noise (here,  $\theta_d = 10\text{cm}$  given by the noise level of the camera).

For each time step  $t$ , the reconstruction module gets as input: the current static background  $S_{t-1} = \{\vec{s}_{t-1}^i\}_{i=1\dots n}$ , the current frame  $F_t = \{\vec{f}_t^i\}_{i=1\dots n}$ , and dynamic objects

points  $D_t \subset F_t$ . These are provided by the tracking module and consist of 3D points that should be excluded from the static scene reconstruction (object hulls in Figure 5(a)– 5(d) are drawn in light green). To handle noise, points annotated with a velocity vector bigger than a certain threshold  $\theta_v$  (here, 3cm which is the variance of the noise) are marked as dynamic (pixels in Figure 5(a)– 5(d) highlighted in dark green) and are excluded from the reconstruction process. As described in Figure 6, the main tasks of this algorithm are to detect those points in the current frame  $F_t$  that improve the static scene (line 12), that define a new static scene point (line 16), and those points that represent movable objects  $O_t$  (line 19). If the distance of a point  $\vec{f}_t^i$  to the corresponding static point  $\vec{s}_{t-1}^i$  is smaller than  $\theta_d$  then it is accumulated to a new static point  $\vec{s}_t^i$  with improved reliability. Otherwise, it has to be determined whether a new static scene point is introduced or a movable object was measured which has to be excluded from the background reconstruction process.

Figure 5 presents the evolution of the background model through a whole sequence. For explanation, frames 17, 70, 96, and 116 are shown. The blue 3D points are the static background points while the orange colored points mark the detected movable objects. It can be seen that the chair was moved and placed in the corner, the door was opened, and the bear was put on the table. All these objects are marked correctly and are not included into the background model which reduces the errors in this model. Also, the points belonging to the chair at their first position are successfully removed from the static scene. Considering the history of the points detected as movable objects, it is even possible to group points together with regard to the point in time they appeared resulting in a model independent object detection method. In the figures, this fact is expressed by different orange notes. E.g., in frame 116 the orange point cloud is split up into two parts meeting convincingly the chair and the bear. The main contribution of this algorithm is the ability to detect regions of movable objects without using specialized object models or classifiers and simultaneously to adapt the background model.

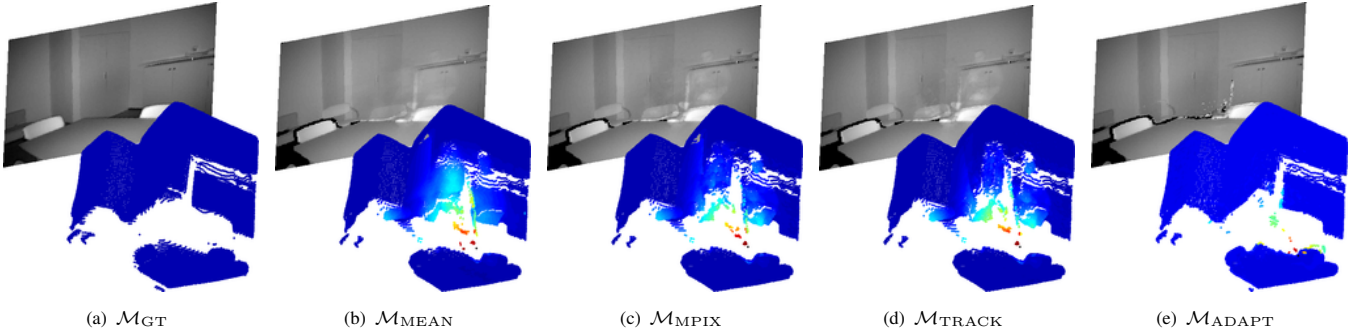


Fig. 7: Results of scene  $\mathcal{S}_{s2,r1}$  for the evaluated algorithms. In the front the reconstructed 3D static scenes and in the back the accordant 2D images can be seen. (a) shows the ground truth. In (b) the reconstruction by simple averaging, in (c) the reconstruction by excluding moving pixels, and in (d) the reconstruction by tracking objects is shown. In the 2D image the wrong reconstruction can be seen as a ghost of the person moving in the scene. (e) shows the result using the here proposed method. The colors encode the error of the model if compared to the ground truth – blue means small and red means big error.

	$\mathcal{S}_{s1,r1}$	$\mathcal{S}_{s1,r2}$	$\mathcal{S}_{s1,r3}$	$\mathcal{S}_{s1,r4}$	$\mathcal{S}_{s1,r5}$	$\mathcal{S}_{s1,r6}$	$\mathcal{S}_{s2,r1}$
$\mathcal{M}_{MEAN}$	103 ± 177	106 ± 204	124 ± 222	157 ± 284	142 ± 278	147 ± 262	95 ± 187
$\mathcal{M}_{MPIX}$	64 ± 121	74 ± 184	79 ± 185	111 ± 216	99 ± 230	95 ± 193	71 ± 155
$\mathcal{M}_{MTRACK}$	71 ± 166	108 ± 209	75 ± 189	97 ± 212	79 ± 308	98 ± 219	84 ± 182
$\mathcal{M}_{ADAPT}$	18 ± 59	19 ± 47	21 ± 61	24 ± 78	24 ± 68	21 ± 55	20 ± 96

	$\mathcal{S}_{s2,r2}$	$\mathcal{S}_{s3,r1}$	$\mathcal{S}_{s3,r2}$	$\mathcal{S}_{s4,r1}$	$\mathcal{S}_{s4,r2}$	$\mathcal{S}_{s4,r3}$	$\mathcal{S}_{s4,r4}$
$\mathcal{M}_{MEAN}$	108 ± 147	89 ± 105	85 ± 183	219 ± 403	321 ± 639	234 ± 451	246 ± 594
$\mathcal{M}_{MPIX}$	80 ± 118	63 ± 145	61 ± 125	163 ± 328	299 ± 635	229 ± 588	229 ± 588
$\mathcal{M}_{MTRACK}$	85 ± 140	71 ± 141	134 ± 712	51 ± 165	74 ± 218	356 ± 677	246 ± 601
$\mathcal{M}_{ADAPT}$	16 ± 37	20 ± 58	22 ± 52	14 ± 26	75 ± 319	18 ± 64	98 ± 404

TABLE I: Evaluation of four reconstruction methods on 14 sequences (mean error ± mean variance). The error shown in the table is computed as mean Euclidean distance over all model points to the corresponding ground truth points. The mean error is given in mm as well as the mean variance.

## VII. RESULTS

The presented experiments are evaluated on challenging scenes with changing background and moving persons in the foreground. The persons partly move very slowly so that they are difficult to determine as non-static scene parts. Furthermore the persons interact with the environment, they move chairs, open and close doors and they rearrange objects in the scenery.

In the following, the proposed system  $\mathcal{M}_{ADAPT}$  is evaluated by comparing the results to the naive approach of only summing up the images and building the mean for each pixel ( $\mathcal{M}_{MEAN}$ ). It is also compared to the neglecting of moving pixels  $\mathcal{M}_{MPIX}$  and last, to  $\mathcal{M}_{TRACK}$  [12] where only dynamic objects are determined through tracking without background model feedback and no distinction is made between static background and static movable objects. All methods are checked against a ground truth static scene model  $\mathcal{M}_{GT}$ , which has been taken without any movable object for each sequence.

To evaluate the proposed algorithms we created 14 sequences  $\mathcal{S}$ , each showing a short scene including a moving person rearranging objects. The sequences can be divided into 4 scenarios, one rearranging Teddy bears  $\mathcal{S}_{s1}$ , one searching a Teddy  $\mathcal{S}_{s2}$ , one tidying up scene  $\mathcal{S}_{s3}$ , and finally opening and closing doors  $\mathcal{S}_{s4}$ . Each run  $i$  of a sequence belonging to one scenario  $j$  is labelled with  $\mathcal{S}_{s_j,r_i}$ .

In Figure 7 the 3D reconstruction results of the evaluated algorithms for scene  $\mathcal{S}_{s2,r1}$  are shown. In the first image 7(a) the ground truth is presented. In Figure 7(b) - 7(c) the reconstruction error gets apparent, as the person slightly gets visible at every position, where the person paused for a

few frames. We even tested to track and exclude the person from the scene reconstruction (see Fig. 7(d)), but without feedback of the static scene. The result is poor compared to the presented approach with feedback of the static scene (Fig. 7(e)).

The results of all sequences are summarized in Table I. The error shown in the table is computed as mean Euclidean distance over all model points to the corresponding ground truth points. They are promising as the mean error of the  $\mathcal{M}_{ADAPT}$  is never above 10cm, but mostly at 2cm. Even in scene  $\mathcal{S}_{s4,r4}$ , where sparse static points in the door can be detected, the result of the proposed method is much more robust than the naive approaches, where the mean error is always above 20cm. The standard deviation for the 3D points of  $\mathcal{M}_{ADAPT}$  averages mostly low as well, which denotes overall stable points. Our method also outperforms the results of  $\mathcal{M}_{TRACK}$ . In Figure 8 three exemplary results are shown. Three images in the bottom left and the image in the background give an impression of the actions in the scenery. The blue points indicate the background, while the orange and red colored points mark the movable objects detected.

Figure 9 gives an impression of the wide variability of detected objects (ranging from several soft toys to chairs and doors) on the salient object regions.

In Figure 10(a) and 10(b) the qualitative comparison of the two tracking approaches is presented. In the birds eye view of the trajectories the red pixel denote the dynamic points and the blue pixel the static points. Figure 10(a) demonstrates the difficulty to detect moving objects in the presence of many dynamic points. In 10(b) the detection of the static

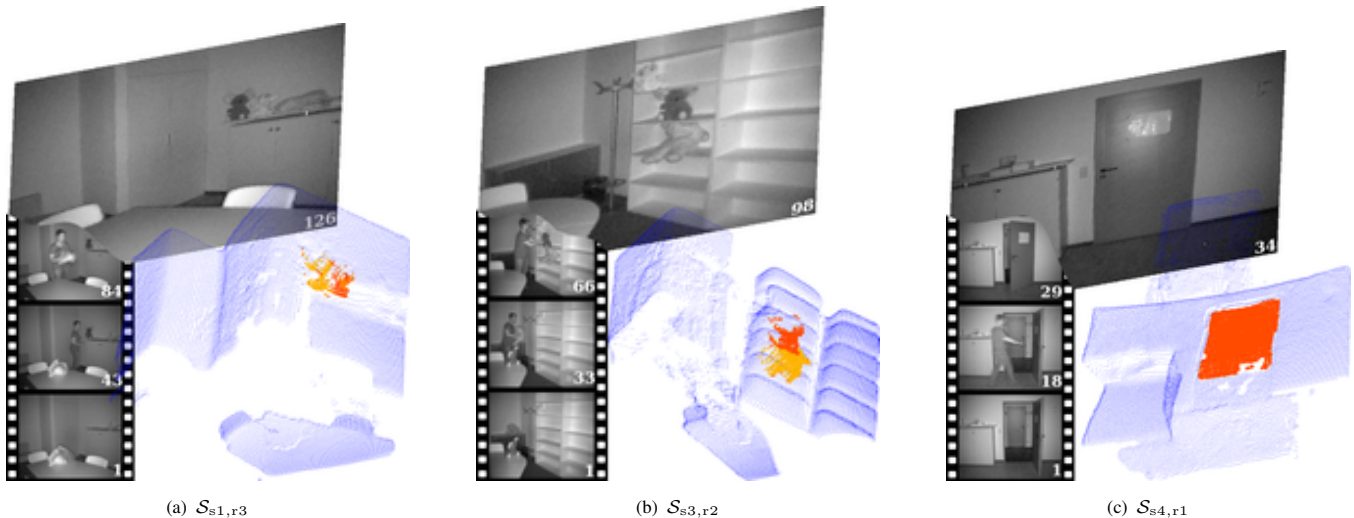


Fig. 8: For three recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.

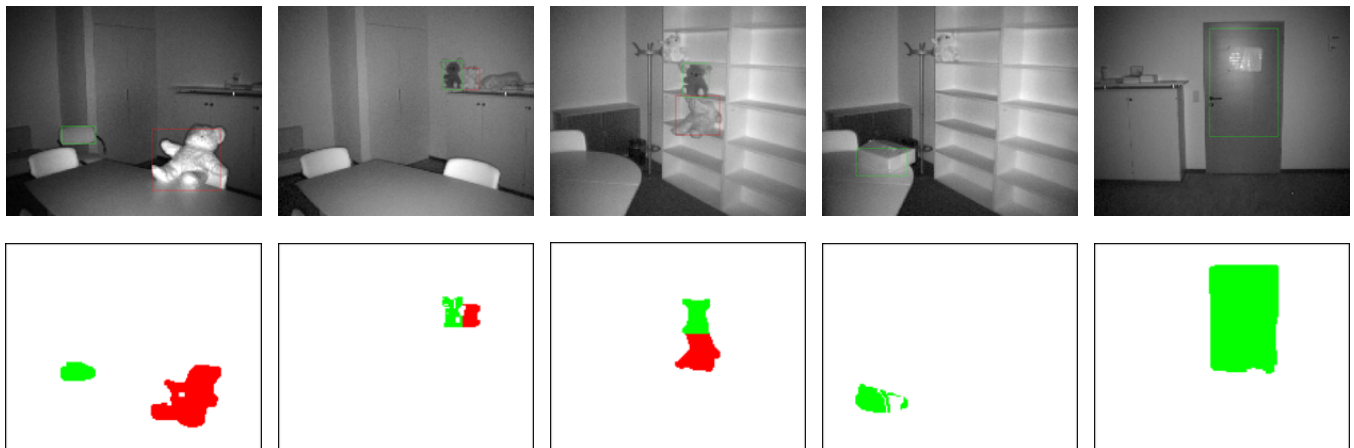


Fig. 9: The images show diverse objects detected by our method. All presented objects have been moved around by the human in the scene. Different colors encode different approaches.

scene reduces the amount of dynamic pixels and therefore, improves the tracking of the real dynamic objects.

Figures 10(c) - 10(l) show the final static scenes and the movable objects of each sequence.

## VIII. CONCLUSION

We presented a combined tracking and reconstruction approach to enable a mobile robot to reconstruct a static scene from a sequence disturbed by moving objects utilizing range and intensity data from a ToF sensor. Assuming a static camera for 2 – 5 seconds length which is typical for a human robot interaction scenario, robust results for such short sequences are provided. The direct connection between tracking and reconstruction at each time step  $t$  accelerates and improves the tracking of dynamic objects as the current background model can be utilized for excluding points from the tracking procedure. Additionally, over the whole sequence the background model is always adapted to the farthest measurement enabling simultaneously a movable object extraction. So far, the emerging structures of the articulated scene model are based on a short observation

interval. In future work knowledge from several intervals should be combined to build up a more complete model of the scenery including presumption of new articulated scene parts, the robot’s ego-motion, and segmented objects for learning scenarios.

## REFERENCES

- [1] I. Lütkebohle, J. Peltason, L. Schillingmann, B. Wrede, S. Wachsmuth, C. Elbrechter, and R. Haschke, “The curious robot – structuring interactive robot learning,” in *ICRA*, 2009, pp. 4156–4162.
- [2] J. Sturm, V. Predeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard, “Learning kinematic models for articulated objects,” in *IJCAI*, 2009, pp. 1851–1856.
- [3] E. Hayman and J.-O. Eklundh, “Statistical background subtraction for a mobile observer,” in *ICCV*, 2003, pp. 67–74.
- [4] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
- [5] I. Cox, S. Hingorani, and S. Rao, “A maximum likelihood stereo algorithm,” *CVIU*, vol. 63, no. 3, 1996.
- [6] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, “Spacetime stereo: A unifying framework for depth from triangulation,” *PAMI*, vol. 27, no. 2, 2005.
- [7] U. Franke, C. Rabe, H. Badino, and S. Gehrig, “6D-vision: Fusion of stereo and motion for robust environment perception,” in *Lecture Notes in Computer Science: DAGM-Symposium*, 2005.



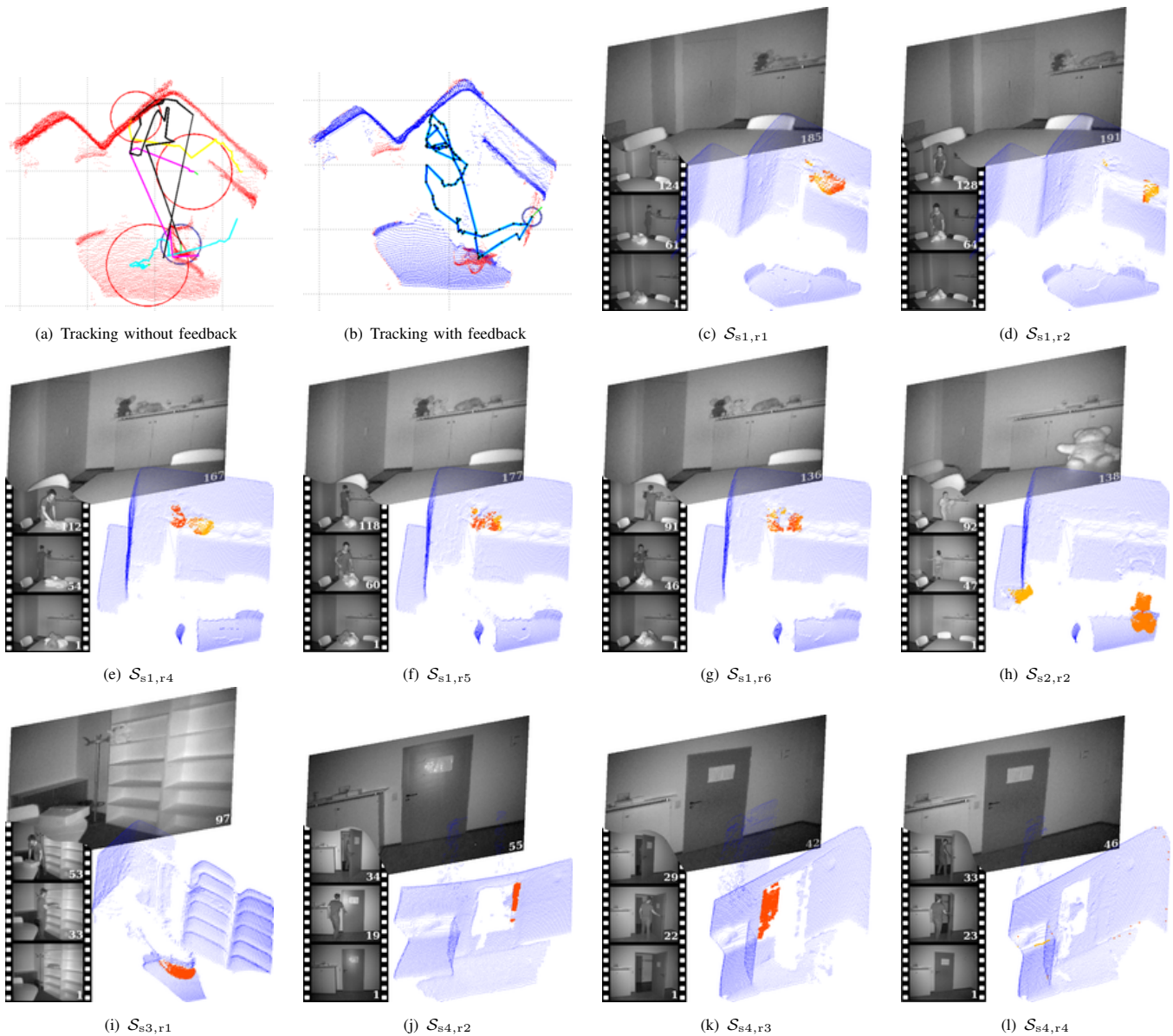


Fig. 10: In (a) the tracking without feedback of the static scene is shown. Because of the amount of dynamic pixels, several objects are found. The different colors denote the trajectory for each object tracked. In (b) the tracking results with feedback are well-defined. In the top-view the red pixel denote the dynamic and the blue ones the static parts of the scene. (c)-(l): For all recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.

- [8] M. Montemerlo, W. Whittaker, and S. Thrun, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *ICRA*, 2002.
- [9] C.-C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *IJRR*, vol. 26, no. 6, 2007.
- [10] J. Weingarten, G. Gruener, and R. Siegwart, "A state-of-the-art 3D sensor for robot navigation," in *IROS*, 2004.
- [11] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck, "Smart pixel - Photometric Mixer Device (PMD) / new system concept of a 3D-imaging-on-a-chip," in *International Conference on Mechatronics and Machine Vision in Practice*, 1998, pp. 259–264.
- [12] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer, "Tracking objects in 6d for reconstructing static scenes," in *CVPR Workshops*, 2008.
- [13] V. Sharma and J. Davis, "Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians," in *ICCV*, 2007.
- [14] B. Möller and S. Posch, "Detection and tracking of moving objects for mosaic image generation," in *Lecture Notes in Computer Science: DAGM-Symposium*, vol. 2191, 2001, pp. 208–215.
- [15] M. Keck, J. Davis, and A. Tyagi, "Tracking mean shift clustered point clouds for 3d surveillance," in *International Workshop on Video Surveillance & Sensor Networks*, 2006, pp. 187–194.
- [16] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 915–926, 2008.
- [17] J. Hois, M. Wünnel, J. A. Bateman, and T. Röfer, "Dialog-based 3D-image recognition using a domain ontology," in *Spatial Cognition V: Reasoning, Action, Interaction*, ser. Lecture Notes in Artificial Intelligence, 2006, no. 4387, pp. 107–126.
- [18] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots – an object based approach," *Robotics and Autonomous Systems*, vol. 55, pp. 359–371, 2007.
- [19] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.
- [20] M. Berthold and D. Hand, *Intelligent Data Analysis*, 2nd ed. Springer, 2003.
- [21] J. Schmidt, C. Wöhler, L. Krüger, T. Gövert, and C. Hermes, "3d scene segmentation and object tracking in multicocular image sequences," in *ICVS*, 2007.