Topical Editor: Prof. Christian Wöhler, 3D RESEARCH **03**, 04 (2010)
10.1007/3DRes.03(2010)04

3DR EXPRESS

Using Articulated Scene Models for Dynamic 3D Scene Analysis in Vista Spaces

Niklas Beuter • Agnes Swadzba • Franz Kummert • Sven Wachsmuth

Received: 29 August 2010 / Revised: 29 September 2010 / Accepted: 10 October 2010
© 3D Research Center and Springer 2010

Abstract In this paper we describe an efficient but detailed new approach to analyze complex dynamic scenes directly in 3D. The arising information is important for mobile robots to solve tasks in the area of household robotics. In our work a mobile robot builds an articulated scene model by observing the environment in the visual field or rather in the so-called vista space. The articulated scene model consists of essential knowledge about the static background, about autonomously moving entities like humans or robots and finally, in contrast to existing approaches, information about articulated parts. These parts describe movable objects like chairs, doors or other tangible entities, which could be moved by an agent. The combination of the static scene, the self-moving entities and the movable objects in one articulated scene model enhances the calculation of each single part. The reconstruction process for parts of the static scene benefits from removal of the dynamic parts and in turn, the moving parts can be extracted more easily through the knowledge about the background. In our experiments we show, that the system delivers simultaneously an accurate static background model, moving persons and movable objects. This information of the articulated scene model enables a mobile robot to detect and keep track of interaction partners, to navigate safely through the environment and finally, to strengthen the interaction with the user through the knowledge about the 3D articulated objects and 3D scene analysis.

Keywords Vista space, Articulated scene model, Mobile robot, Person tracking, 3D background modeling

1. Introduction

Embodied agents, both humans and mobile robots, have to perceive, to analyze and to segment observed scenery into meaningful parts to deal with and communicate about the unknown and dynamic environment. In this paper we present a 3D scene analysis approach, which enables mobile robots to solve such problems by gathering broad knowledge about their environment only by observation of the scenery. Because of the broad area of requirements the robot needs information about different parts of the world. First, the robot has to detect and track humans as possible interaction partners or moving obstacles to avoid possible collisions. Second, the static scene parts like walls, cupboards or tables have to be segmented to give a broad knowledge about the room structure for e. g. navigation purposes[45] or room classification[41]. In contrast to other typical background modeling approaches[36,19,26], our suggestion is to distinguish as well between static objects and objects like chairs, teddy bears or other smaller objects that can be moved by an agent[42]. Instead of building a complex ontology of human environments to describe which parts may be moving or could belong to the static background and equipping the robot with strong detectors for each possibility, we propose to learn an articulated scene model on the basis of scene observation. This bottom-up learning of spatial awareness enables a mobile robot to extract essential knowledge about the environment, which is achieved only by observation. The articulated scene model is composed of the following three scene parts.

Definition 1. (Articulated scene model):

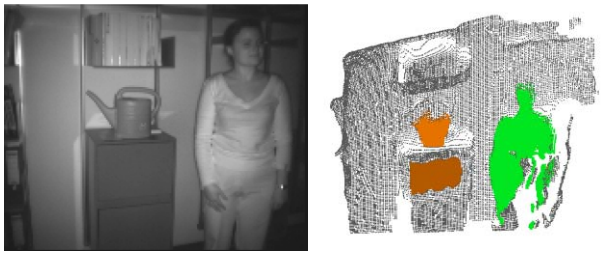
- *Static scene (Never changing parts)*
- *Moving entities (e. g. humans or robots)*
- *Movable objects (e. g. chairs, doors)*

This model is updated in one single and simultaneous computation. Fig. 1 is meant to give an example. On the left the accordant frame of the scene is presented and on the right an example of a 3D articulated scene model is shown. Colored in black is the static background, in orange

Niklas Beuter (✉) • Agnes Swadzba • Franz Kummert • Sven Wachsmuth
Bielefeld University 33615 Bielefeld , NRW GERMANY
e-mail: nbeuter@uni-bielefeld.de

and brown are two articulated objects and in green the actual tracked human is displayed.

Usually, the observation of an environment refers to the large-scale-space[22,27], where a main property is the necessity of locomotion to perceive the space, which could be, e.g., a complete flat or apartment. In the proposed system we apply the observation on the so-called vista space, which describes the visual field only by slightly moving the gaze.



(a) 2D amplitude image (b) 3D point cloud

Fig. 1 In the left image the frame of an example sequence is shown. In (b) two detected articulated scene parts are shown (cupboard door, water can) in red and orange, which emerge after a few seconds of observation, if an agent moves the specific object. The gray 3D points belong to the background and the green points to the current tracked person.

Definition 2. (Vista space):
The vista space is a part of the world, which can be viewed at the same moment, only be slightly moving the gaze.

This means that the system relies on the perception of a single room or parts of a room and that the short observation time limits the number of available frames. By the use of the vista space we can derive the assumption that the farthest measurement in the scene describes the background. If an object appears in front of previously seen static parts, we can assume a moved object, while upcoming observations of more distant points indicate a removed object.

Assumption 1. (Vista space assumption):
The farthest measurement in the scene describes the background.

As vista space models deliver complementary information to large-scale space models the combination of both model types into a common representation, e.g., using the Hybrid Spatial Semantic Hierarchy (HSSH) of Beeson et. al[1] will form the foundation for modeling spatial knowledge of the entire environment an agent interacts with. However, in this paper we focus on the analysis of the vista space as we are going to integrate the system in the home-tour scenario[7], in which a user guides the robot around in his flat and particular vista space situations arise.

The robot needs a meaningful sensory input to perceive the environment, which in our case is achieved by using a time-of-flight 3D camera. The 3D data is extended with additional 3D velocities using optical flow. The use of a 3D sensor translates the problem to an inherent 3D interpretation task.

Our proposed system builds the articulated scene model only by observing the 3D scenery for a few seconds, thereby segmenting the environment into different parts and incorporating the already learned knowledge. The humans in the observed scene are detected by consideration of velocity information and a weak object model suitable for many different kinds of objects. The human is tracked by a hybrid particle filter with mean shift, which enables the robot to keep track of the movements of the human. The calculated trajectories supply a broad knowledge about the typical movement areas in the scene and additionally, the robot gets the required positions of possible interaction partners.

In contrast to other background modeling strategies, the articulated parts of the scene are separated from the static scene, which are normally incorporated again into the background model after the objects become static again. Usually, even with a strong detector the articulated objects are hard to detect as they could have any shape or size. Here, the articulated parts are detected through the vista space assumption.

The static scene is composed of the remaining parts after excluding the persons and the movable objects. Through the exclusion of dynamic parts the static scene is very reliable for navigation or scene classification, as many potentially changing parts have been already removed.

However, the main advantages of the proposed system are based on the parallelism and the generality of the detection of the different parts of the articulated scene model. Through the detection and exclusion of moving persons and movable objects the building of the static scene is much more robust. On the other hand the knowledge about the static scene enhances the detection of humans as the static background could be subtracted and the detection can be limited to dynamic parts of the current observation. The static scene again is used in the assumption of the vista space to detect the articulated scene parts. Different to existing approaches, movable objects can be detected without the explicit detection of a movement of the particular object, but through the knowledge about the static scene and the information from observation.

The contribution of the proposed model is a solid basement of information, which could be used by many other applications as input. In the following, we want to present some ideas or possible applications for the articulated scene model. The possibilities are comprehensive as the model is a good starting point for several learning or interaction scenarios. As mentioned before, the tracked persons or moving objects could be directly associated as interaction partners. On the other hand, the information about their movements can be used as data for typical movement areas or pathways, which could be used for navigation purposes of the robot. The articulated parts apparently enable the robot to recognize objects, which are handled by the human or more simply, which objects are movable. This knowledge could be utilized in a tabletop-learning scenario, where each object put onto the table could be easily recognized as a new object independent from its topology or appearance. Again, using the whole information about the recognized objects and the appearance and disappearance areas we get an idea about the action spaces of these objects. In the case of a door, we could see the articulation or the opening range of the door as an action area. Several other scenarios are supposable, but as the main contribution of this paper is a solid basement of knowledge for a mobile robot we skip

further suggestions how to use the articulated scene model in a specific application.

The paper is structured as follows. First, related work is presented in section 2 to give an overview of other work done in the different fields covered by this paper. The proposed system in general is described in the subsequent section 3. The preprocessing of the sensor data is explained in section 4 and the computation of the extended optical flow in the subsequent section 5. The detection and tracking of moving entities is described in section 6, followed by the description how to build the static parts and how to detect the movable parts of the articulated scene model in section 7. In the end, we explain our experiments and we show the results of our algorithm on several self-created data sets in section 8.

2. Related Work

Research on dealing with dynamic scenes has become more and more important since the manual analysis of the huge amount of video data provided by video surveillance is not suitable any more. Diverse methods have been developed to model the background that can be subtracted from the current image to extract the moving foreground. The approaches range from classical Gaussian Mixture Models (GMMs)[36] to the use of codebooks[19] modeling the pixels either separately from each other or incorporating nearby pixels using subspaces[26]. For a lot of approaches a static background is mandatory however Sheikh and Shah introduced an approach that is able to cope with uniformly moving background like a river[35]. The work from Brostow and Essa[4] describes the other way around. They propose to observe the foreground and to analyze the motion of dynamic objects to decompose a scene from a single view as multiple layers. By extending the single view to multiple views Guan et al. [12] recover static 3D shapes of static occlusions by observing the human motion shapes. Multiple views including depth-sensors are able to reconstruct reliable 3D scenes, if the camera setup is well calibrated[13]. The problem of a moving camera has to be considered, if we transfer approaches for detecting moving regions developed in a static surveillance scenario to a robotic scenario. This can be done directly by an ego-motion compensation[34], by visual navigation[10] or by detecting moving objects through inconsistencies in a scene motion field arising from a optical flow computation[20]. Another problem in robotics scenarios is the short observation time and the unknown environment so that a previous training of the background is not possible. Therefore, Hayman and Eklundh[14] developed a Bayesian model for incorporating the possibility that the background has not yet been uncovered.

Besides from moving persons also movable objects are interesting for a robot. Movable objects are characterized by occasional relocation and longer static periods. In classical background subtraction approaches such objects will be integrated into the background model after relocation thus cannot be detected anymore[29]. Sanders et al. [31] try to solve this problem by integrating pixel information over time. The pixel history is clustered to temporal coherent clusters, the so-called temporal signatures, which allow detecting quasi-static objects under the condition of these objects having arrived and departed from the scene. As movable objects belonging to the class of scene structuring elements like a chair are of special interest for a robot some approaches try to find such scene elements through analyzing the human activity instead of detecting them directly. For example, trajectories can be segmented to actions using Hidden Markov Models (HMMs)[29] concluding that the location of an action points to an object associated with an action like, e. g., “sitting down” is coupled with a chair. Alternatively, clustering of motion histograms computed per scene cell allows an image segmentation providing interesting indoor scene regions like a sofa[9]. The analysis of trajectories of moving objects can reveal – besides image regions that correspond to scene elements – general semantic regions like junctions or paths that do not match a specific movable object. Analyzing person trajectories in indoor rooms could reveal semantic regions like a grouping of table and chairs[21]. Analyzing person activities and car trajectories in outdoor environments could provide models of roads and paths[45], “walkable” ground surfaces[3], or routes, paths, and junctions[24]. A detailed review of further methods for understanding scene activity is given in[6].

In the case of detecting movable objects, e. g. , a door, which motion is caused by a human manipulation[31] trajectories of such objects reveal their possible articulation. Inspired by articulated body models, Sturm and colleagues[38] developed techniques for learning kinematic models of scene elements like table or drawer. As tracking of such objects is a challenging problem they bypass it in their paper through attaching markers to test objects. In their last paper[37] they have presented an automatic tracking of a planar surface from a cupboard door or a drawer front for observation situations restricted to a close-up view of the surface.

Our articulated scene model aims to combine background modeling with detection of semantic scene elements. As we focus on the modeling of dynamic 3D scenes the assumption that static measurements which are furthest away determine the scene background allows an elegant way to model the background especially in robotic scenarios where observation times are short. Subtracting the background in 3D reveals directly quasi-static/articulated objects without special requirements like an object has to arrive and depart[31] and independent from their shape or size or the human activity connected to them. Detecting arbitrary articulated scene elements using human activity requires recognition abilities of a lot of different daily-life activities, which means that a huge database of all possible actions is needed for training. Our approach provides for 3D data a bypass to this exhaustive learning problem.

3. System Overview

The robot’s purpose is to interact with the human and to work with him in the same environment, but the environment is naturally not static and the human moving in front of the robot is inhibiting the background modeling process. Therefore, the robot should acquire knowledge

about its surrounding by detecting and tracking moving objects, modeling the static background without these persons and perceiving scene changes in the vista space. In the process the robot observes its environment passively, which means the robot camera stays static for a few seconds to gather information before the robot changes its view and observes the next vista space.

The algorithm is designed to calculate an articulated scene model M for each of the vista spaces (see fig. 2). The model consists of the dynamic parts *D*, the static background S and the observed articulated scene parts *O*. The model for one vista space is updated as long as the robot does not change its view. The model M_{t-1} is updated by propagating it to the next frame at time *t*. In each frame the following processes are accomplished to update the model:

1. Model propagation: The model M_{t-1} from the previous frame is propagated to the current frame
2. Perception & Preprocessing: The actual sensor input is preprocessed and annotated with velocities
3. Entity Tracking: Moving objects are detected and tracked to exclude them from the static scene
4. Scene Modeling: The background and the movable objects are adapted

The preprocessing cares for the 3D data smoothing and velocity computation V_t based on optical flow resulting in 6D data as sensor input for frame t. The next step is to detect and track the moving parts, named *as Entity Tracking*. Thereby, the detection and tracking of moving persons is supported by the knowledge of the actual static scene S_{t-1} generated out of all previous frames and vice versa.

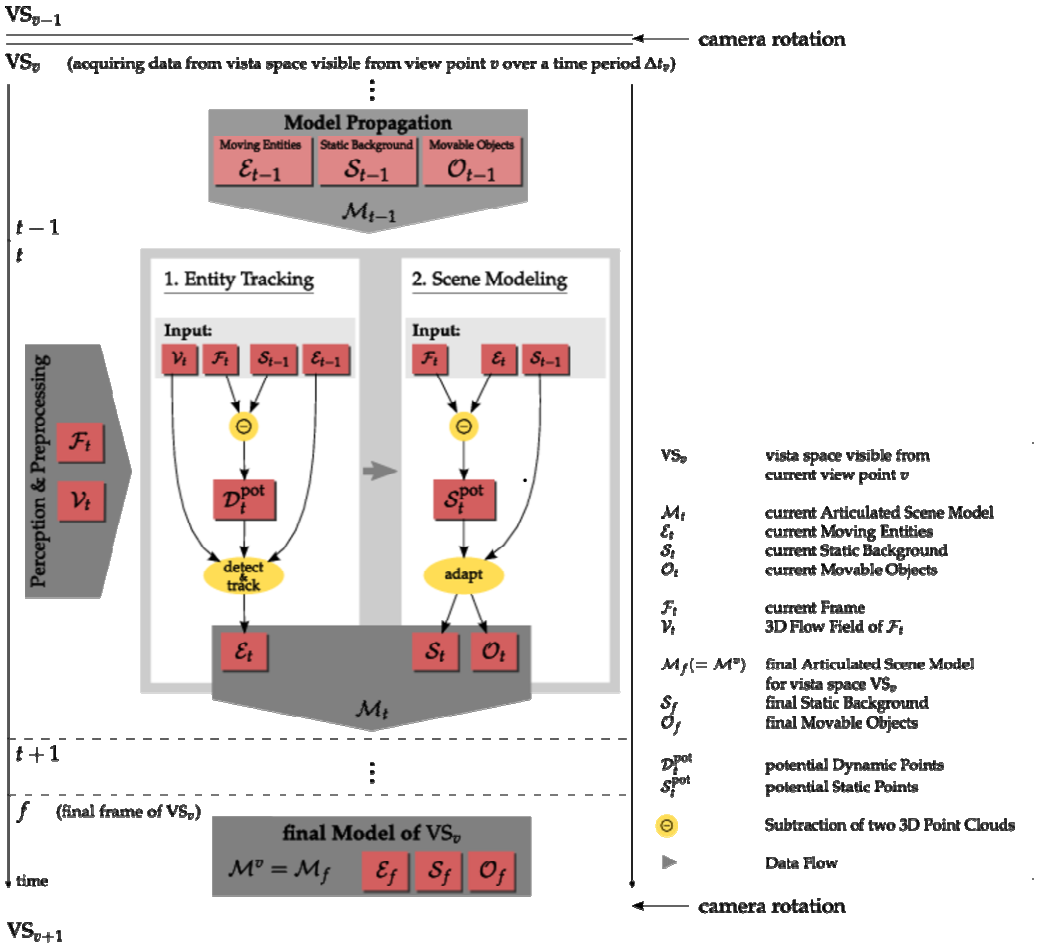


Fig. 2: The articulated scene model is calculated for each vista space. The model is updated from frame to frame by observing the scenery. Utilizing the model from the previous frame and the sensor data from the current frame the updated model can be calculated by two steps. First, the entity tracking detects and tracks moving objects by shifting a cylindrical model through the potential dynamic points D_t^{pot}. The potential points are all points, which are not conform to the known static background. Second, the static scene and the articulated objects are adapted. Therefore, all found moving objects are subtracted and the produced potential static points S_t^{pot} are analyzed with the vista space assumption to separate movable objects from the updated static scene.

In a first step, the known static scene points n from the previous frame

$$S_{t-1} = \left\{ \bar{s}_t^i \right\}_{i=1\ldots n} \quad (1)$$

are subtracted from the current scene

$$F_t = \left\{ f_t^i \right\}_{i=1\ldots n} \quad (2)$$

The remaining potential dynamic points

$$D_t^{pot} = F_t - S_{t-1} \quad (3)$$

are annotated with the velocity data V_t. Based on the potential dynamic points D_t^{pot} new objects are detected.

Using a clustering algorithm and a simple cylindrical object model, the moving objects are found and subsequently tracked with a hybrid particle filter with mean shift. The potential points

$$\varepsilon_t \subset D_t^{pot} \quad (4)$$

which belong to a dynamic object are passed to the current articulated scene model M_t.

In the *scene-modeling* step these points ε_t are subtracted from the actual frame F_t to identify the potential static points

$$S_t^{pot} = F_t - \varepsilon_t \quad (5)$$

in the current frame. By applying the vista space assumption and utilizing the knowledge S_{t-1} from the last frame the movable objects O_t that form the articulated scene parts can be detected and the static background S_t can be updated, simultaneously. Both are passed to the current articulated scene model M_t, which is propagated again to the next frame.

The updating of the vista space ends if the robot changes its view and during the motion of the camera from one vista space to another the model computation is stopped. At this moment the outcome from the articulated scene parts O_t are all the areas where a movable object is newly detected by the vista space assumption. From the moment the robot observes a new vista space the building of the next articulated scene model begins. By incorporating the motion of the robot the vista spaces can be merged to build a global knowledge base. Here, we utilize the motion information from a laser-based SLAM approach[45]. The typical observation time for one vista space is about 15 to 20 seconds.

4 Preprocessing

Our system uses the Swissranger SR3000 provided by Swiss Center for Electronics and Microtechnology (CSEM) [45] delivering a matrix of distance measurements independent from texture and lighting conditions. It consists of 176×144 CMOS pixel sensors, which are able to determine actively the distance between the optical center of the camera and the real 3D world point via measuring the time-of-flight of a near-infrared signal. Besides the distance value matrix (Fig. 3(b)), the camera provides per frame a matrix containing amplitude values (Fig. 3(a)). The amplitude value indicates the amplitude of the reflected near-infrared signal received by the sensor and implies therefore the amount of light reflected by a world point. A small amplitude corresponds to a small amount of light reflected and therefore indicates a weak signal.

Several researchers have already developed preprocessing and calibration techniques dealing with noise arising from the different reflectance properties and characteristics of the ToF cameras, like additional infra-red light in the scene, and measurement errors at edges (so-called "flying pixels"). Schiller[32] proposed an automatic calibration of the entire 3D ToF signals using a bunch of different cameras. Color information is also used by Huhle[18] for outlier detection and smoothing using Non-local Means filter[5]. Smoothing techniques relying only on the ToF data are amplitude thresholding with a fix value[25], removing of "flying pixels" at edges via detecting iteratively geometric outliers taking into account the 2D neighborhood[17], and correcting the amplitude values using distance values and vice versa[28]. In this paper we applied preprocessing techniques proposed in [40].

The distance image is smoothed with a distance-adaptive median filter, which uses for each pixel a different mask size (e.g. 3×3, 5×5, or 7×7) depending on the distance value of the pixel. Generally, pixels with larger distance value are filtered with smaller filter masks, and vice versa, so that significant structures at large distances are not blurred, and at the same time, noisy surfaces at small distances can be smoothed. As the amplitude value refers to the quality of the distance measurement, points with a small amplitude value are removed from the final 3D point cloud. The threshold needed adapts automatically to different reflectance properties in different scenes, as it is a fraction of the mean amplitude value per frame. Further, edge points (so-called "flying pixels") arising in the case where light from the fore- and the background hits the same pixel simultaneously are rejected if the amount of near neighbors in the 2D neighborhood is insufficient. Last, 3D coordinates are generated out of the distances with regard to a 3D camera coordinate system.

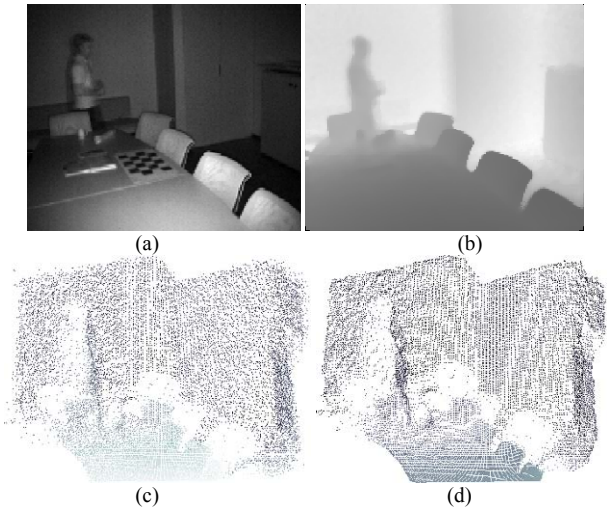


Fig. 3 Raw data acquired from the Time-of-Flight sensor: (a) amplitude image, (b) distance image, (c) unpreprocessed 3D point cloud, and (d) preprocessed 3D point cloud.

With the assumption of ideal perspective projection, the known position of the principal point, pixel sizes, and focal length, the 3D coordinates can be computed from the distances via ray proportions in triangles. As a result the computed 3D points are organized regularly in a 2D matrix. Fig. 3(a) to 3(c) show a frame of the 3D ToF camera consisting of an amplitude image, a distance image, and the raw 3D point cloud. Fig. 3(d) presents the resulting 3D point cloud after applying the described preprocessing techniques.

In order to distinguish between static parts of the scene and moving persons or objects the motion in the 3D point cloud has to be determined. In the following an image-based method for motion computation is presented which

can be applied here easily by treating the point cloud as planar depth maps or images[39].

5. Motion Computing using Optical Flow

A widely used technique is to compute the dense Optical Flow using each 2D image pixel. The optical flow is the distribution of apparent velocity of moving brightness patterns in an image and arises both from the relative objects' and the viewer's motion [11]. The flow of a constant brightness profile

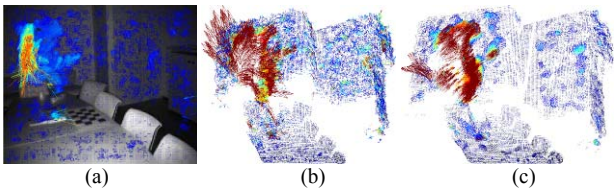
$$\begin{aligned} I(x,y,t) &= I(x+dx, y+dy, t+dt) \\ &= I(x+v_x \cdot dt, y+v_y \cdot dt, t+dt) \\ &= I(x+v_x \cdot dt, y+v_y \cdot dt, t+dt) \end{aligned} \quad (6)$$

$$\frac{\partial I}{\partial x} \cdot v_x + \frac{\partial I}{\partial y} \cdot v_y = -\frac{\partial I}{\partial t} \quad (7)$$

is described by the constant velocity vector $\vec{v}_{2D} = (v_x, v_y)^T$. Usually, the estimation of optical flow is founded on differential methods. They can be classified into global strategies, which attempt to minimize a global energy functional[15] and local methods that optimize some local energy-like expression. A prominent algorithm developed by Lucas and Kanade[23] uses the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. They assume the optical flow to be constant within a certain neighborhood N which allows solving the Optical Flow Constraint Eq. 7 via least square minimization. Here, we have used a hierarchical implementation of Lucas's and Kanade's algorithm written by Sohaib Khan (http://www.cs.ucf.edu/~khan/, http://server.cs.ucf.edu/~vision/source.html).

As the Swissranger camera provides normal 2D intensity images based on the amplitude values it is possible to reduce the 3D correspondence problem to a 2D correspondence problem and to compute the optical flow for each frame of a sequence of ToF images $(F_1, F_2, ...)$ based on data of two consecutive frames (F_i, F_{i-1}). Each pixel of frame F_i is annotated with a 2D velocity vector $\vec{v}_{2D} = (v_x, v_y)^T$ as shown in Fig. 4(a), which results into pixel correspondences between frame F_i and frame F_{i-1}. As 3D information is available for each pixel these pixel correspondences can be directly transformed into 3D point correspondences $\left(\vec{p}_k^{\,i}, \vec{p}_l^{\,i-1}\right)$, which can be used to compute 3D velocities $v_{3D} = \left(v_x, v_y, v_z\right)^T = \vec{p}_k^{\,i} - \vec{p}_l^{\,i-1}$. Fig. 4(b) presents a 3D point cloud of one frame of a test sequence annotated with 3D velocity vectors. The processing from 2D optical flow on 2D images to real 3D velocities is supported by the used hardware. As the Swissranger camera provides good distance measurements velocities with reliable values especially in the z component can be determined. This is usually not suitable for many other camera setups like stereo rigs or multi-camera systems. The velocity annotated 3D point cloud results in 6D data.

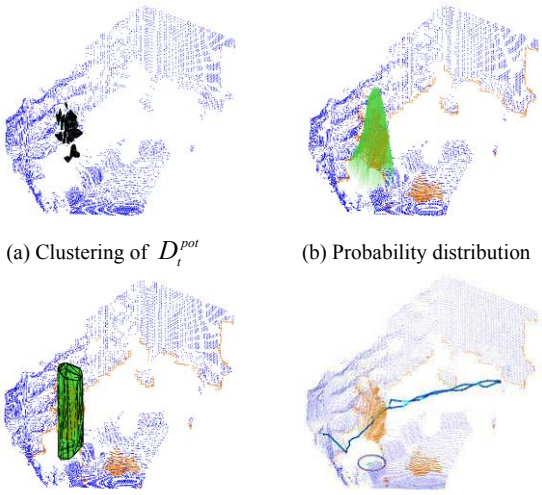
Due to the low resolution of the camera and inaccuracies of the optical flow erroneous velocity vectors at changing depth steps are computed. To get rid of those outliers a 5 5 median filter is applied separately to the three components v_x, v_y, v_z of the flow vector $\vec{v}_{3D}$. In Fig. 4(c) the smoothed result of the 3D velocity field of Fig. 4(b) can be seen.



(a) (b) (c)

Fig. 4 Velocity processing with the optical flow method: (a) 2D velocity vectors (b) 3D velocity vectors from combining 2D velocities and point correspondences in consecutive images, (c) the latter smoothed component wise by a median filter. Each 3rd velocity vector is displayed and color coded with respect to its length: red denoting a big motion vector and blue a small one.

6. Detection and Tracking of Dynamic Objects



(a) Clustering of D_t^{pot} (b) Probability distribution

(c) Found object (d) Trajectory

Fig. 5 The images explain the tracking algorithm. The blue points belong to the static scene S_{t-1}. The dynamic pixels P_t are plotted in orange. (a) At a first stage the dynamic points are clustered, generating small motion attributed regions. (b) The objects are detected and tracked using the observation function (see Eq. **Error! Reference source not found.**). 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) The resulting object trajectory is plotted in cyan. The blue circle contains the object at the actual position.

The dynamic scene analysis involves the detection and tracking of moving objects, which on the one hand enhances the segmentation of the different scene parts and which is on the other hand useful for the understanding of the scene as the trajectories of the objects give a broad picture of the movements in the actual vista space.

Using the 3D point cloud and the annotated 3D velocities, we can simplify the scene by applying a 6D hierarchical clustering technique. The segmentation is enhanced through the incorporation of the velocity information in the early clustering stage, because it enables the segmentation of neighboring objects, like a person walking in front of a wall. The first step is to span small contiguous regions in the cloud of the 6D points, based on features for spatial

proximity and homogeneity of the velocities. We apply a hierarchical clustering using the complete linkage algorithm[2], which, choosing a small branching factor in the hierarchical tree, deliberately over-segments the scene, generating many small motion-attributed clusters[33] (see Fig. 5(a)). Each calculated cluster is annotated with the 2D position of its centroid projected on the ground plane, a weight factor accordant to the number of included points and the mean velocity of all these points.

From here on, persons and objects are represented by an upright cylinder of variable radius, which is a suitable model for the moving entities in our scenarios (here, humans). The object hypothesis *h(a)* is characterized by a five dimensional parameter vector

$$a = [x, y, r, v_\theta, v_r] \tag{8}$$

where x and y are the centroid on the ground plane, r the radius and v_θ the direction and v_r the magnitude of the velocity of the cylinder.

The next step is the detection of the moving objects. Here, the cylindrical model is advantageous for the detection as the velocity computation is noisy and many points between the found clusters have different velocities. This means, that the hypothesis covers mostly the full moving object even if the velocity data is noisy. Thereby, the cylinder does not need to touch the ground or to satisfy any specific height, because all clusters are projected on the ground plane. This is especially useful if the moving object is only partially seen. The detection only needs a few clusters denoting a moving object and afterwards, using the cylindrical model, all clusters that are lying in the cylinder are added to the moving object.

To generate a hypothesis, the cylinder is shifted through the small clusters searching for meaningful collections of clusters with similar velocities. Grouping close clusters together, a hypothesis is found if the weight of all clusters together is higher than a certain threshold. Here, 20 close points moving in a similar direction are sufficient. Thereby, the cylinder has the expansion from the lowest to the highest detected moving cluster. The radius is weighted using a Gaussian with the mean at typical person dimension. Afterwards, each cluster integrated into a hypothesis is marked as an already found object to ensure each cluster is used only for one hypothesis.

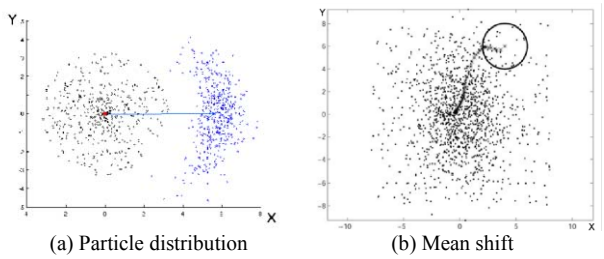
All found hypothesis are additionally annotated with an id to identify them over the observation time. The detection by moving needs the object to move at least one time and afterwards, it is capable to track the object even if it is not moving anymore.

If a potentially moving object has never been seen moving so far, a stronger detection algorithm is needed. As we have different moving objects like persons, robots, cleaning machines or other self-moving objects we would need several detectors and classifiers for each group. Here, we only utilize a human classifier, because most of the attending moving objects in the Home-Tour-Scenario are humans and the other objects are mostly detected with the previously mentioned clustering algorithm. As a human classifier we utilize the Histograms of Oriented Gradients detector[8], which uses the gradient features of local bins consolidated in one histogram. A trained support vector machine differentiates between humans and non-humans, delivering rectangular regions including persons. In the calculated 2D area we cluster the included points with a k-means clustering with k=2 to separate fore- and background. The cylinder is fitted into the 3D foreground and the hypothesis is added to initial detection set.

All extracted hypotheses from the current frame are merged with the ones tracked from the previous frame resulting in one hypotheses matrix for each frame.

The tracking of the hypotheses is calculated like follows. The K hypotheses in the current frame t are tracked based on the position, velocity and size of each hypothesis in the previous frame $h_k^{t-1}(a)$, utilized in a hybrid kernel particle filter with mean shift[33]. The particle filter creates a set of new hypotheses $s_k^t(h)$ for each tracked object, called particles, and distributes them with a first order motion model mixed with a random Gaussian noise (see Fig. 6(a)).

This distribution of particles covers the potential movement of most moving objects as it follows linear and random movement.



(a) Particle distribution (b) Mean shift

Fig. 6 (a) The particle distribution follows a motion model and a random distribution to cover all possible motions of a human. The figure shows the distribution of the particles in the XY plane. The movement of the object is in positive X direction. The particles are distributed in the accordant direction and for random movements as well. (b) The distributed particles are weighted and then shifted with mean shift to recover the best possible object position. (Here, shown for a random distribution)

In order to identify the new position of each hypothesis the particles are rated with an underlying observation. The observation is based on the relative position, relative velocity and weight of all clusters within the cylinder of each hypothesis weighted with Gaussian kernels.

$$\rho(s_k) = K_r(s_k) \sum_{l \in s_k} K_d(l, s_k) K_v(l, s_k) \tag{9}$$

$$K_r(s_k) = \exp\left(-\frac{r(s_k)^2}{2H^2_{r,\min}}\right) - \exp\left(-\frac{r(s_k)^2}{2H^2_{r,\max}}\right) \tag{10}$$

$$K_d(l, s_k) = \exp\left(-\frac{\|d(l) - d(s_k)\|^2}{2H_d^2}\right) \tag{11}$$

$$K_v(I, s_k) = \exp\left(-\frac{\|v(l) - v(s_k)\|^2}{2H_v^2}\right) \tag{12}$$

With $K_r(s_k)$ keeping the radius in a realistic range, $K_d(l, s_k)$ reducing the importance of clusters further away from the cylinder center, and $K_v(I, s_k)$ masking out clusters having differing velocities. The functions r(·), d(·), and v(·) extract the radius, the 2D position on the ground plane and the velocity of a cluster *l* or a hypothesis s_k. The kernel widths H are determined empirically. Eq. **Error! Reference source not found.** is also called the observation function

$\rho(s_k)$ of the particle filter. The outcome is a density approximation based on the object hypothesis and the attributes of the appending clusters, with the maxima corresponding to the actual objects (Fig. 5(b)).

Several mean shift iterations refine the particles to concentrate at the local maxima of the distribution, which decreases the needed amount of particles (see Fig. 6(b)). The combination of mean shift and particle filter is ideal to add the strengths of both parts in one algorithm. Mean shift sometimes sticks in local minima, which could be resolved due to the sampling of the particles. The particle filter needs many particles to estimate the underlying density function, which can be avoided through the combination with mean shift. Individual particles selected from these best modes of the distribution represent the objects found in the current frame (Fig. 5(c)). For each tracked object hypothesis, all 3D points associated with this object are back projected in the 2D amplitude image and used for computing a 2D convex hull of the tracked object. All points within this 2D polygon are marked as non-static points and are finally excluded from the reconstruction step.

7. Adaptive Background Modeling

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

Fig. 7 Algorithm per time step t for background adaptation and movable object detection.

So far we proposed methods to distinguish between static and moving parts in a scene. In the following, the calculated moving objects are extracted and the static parts of the observation are analyzed. By applying the vista space assumption and utilizing the knowledge from the last frame the movable objects that form the articulated scene parts can be detected and the static background can be updated, simultaneously. The basis of the vista space assumption that the most distant measurement in the current view describes the background has to be expanded due to noise of the 3D sensor. Therefore, we introduce a threshold θ_d above that a change in the distance is significant and do not arise from noise (here, $\theta_d = 10cm$ given by the noise level of the camera).

The algorithm presented in Fig. 7 is applied to each time step of the observation to calculate the updated static scene $S_t = \{\vec{s}_t^{\,i}\}$ and the movable objects O_t . Therefore, the algorithm uses as input the current frame $F_t = \{\vec{f}_t^{\,i}\}$ and the last known static scene $S_{t-1} = \{\vec{s}_{t-1}^{\,i}\}$ and the dynamic clusters ε_t from the previous frame. The dynamic clusters contain the 3D points from the moving objects of the tracking module. These points are removed, before the update process takes place.

The static scene is updated in line 11, if the difference of a known static point $\vec{s}_{t-1}^{\,i}$ to the actual frame point $\vec{f}_t^{\,i}$ is below the sensor noise level θ_d . Then, the static point and the current point are accumulated to a new static point $\vec{s}_t^{\,i}$ with improved reliability. Otherwise, it has to be determined if a new static scene point is detected in line 16 or the point belongs to a movable object in line 19. The vista space assumption is used to identify the matching case. All points belonging to movable objects are saved in a separate list, where the time of detection and the number of times the points has been seen are considered. Clustering these points in space and time the different objects can be separated. Consequently, objects can only be separated if they appear at a different point in time or at least at different places.

8. Results

The evaluation of an articulated scene model does not follow typical standard reports, as it is not feasible to build a complete ground truth model. Hence, we split up the different parts of the model and we compared the static scene to a ground truth model and to some simple background modeling techniques to give quantitative results. In the following, the proposed system 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 (M_{MEAN}). It is also compared to the neglecting of moving pixels M_{MPIX} and last, to M_{TRACK} [39] 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 M_{GT} , which has been taken without any movable or moving object for each sequence. The articulated parts and the trajectories of the moving objects are presented qualitatively in illustrations.

The underlying data sets *S* are self-created and they show different challenging dynamic scenes. The human shows different moving behaviors or stops moving, which makes it difficult to detect him as not static. Furthermore, the human interacts with the environment as he cleans up S_{S3} , moves chairs, searches a teddy bear S_{s2} , opens and closes doors S_{S4} and rearranges teddy bears S_{S1} , water cans S_{S5} and plants S_{S6} . Each run i of a sequence belonging to one scenario j is labeled with $S_{sj,ri}$.

	$S_{s1,r1}$	$S_{s1,r2}$	$S_{s1,r3}$	$S_{s1,r4}$	$S_{s1,r5}$	$S_{s1,r6}$
M_{MEAN}	103±177	106±204	124±222	157±284	142±278	147±262
M_{MPIX}	64±121	74±184	79±185	111±216	99±230	95±193
M_{TRACK}	71±166	108±209	75±189	97±212	79±308	98±219
M_{ADAPT}	18±59	19±47	21±61	24±78	24±68	21±55

	$S_{s2,r1}$	$S_{s2,r2}$	$S_{s3,r1}$	$S_{s3,r2}$	$S_{s4,r1}$	$S_{s4,r2}$
M_{MEAN}	95±187	108±147	89±105	85±183	219±403	321±639
M_{MPIX}	71±155	80±118	63±145	61±125	163±328	299±635
M_{TRACK}	84±182	85±140	71±141	134±712	51±165	74±218
M_{ADAPT}	20±96	16±37	20±58	22±52	14±26	75±319

	$S_{s2,r1}$	$S_{s2,r2}$	$S_{s3,r1}$	$S_{s3,r2}$	$S_{s4,r1}$	$S_{s4,r2}$
M_{MEAN}	95±187	108±147	89±105	85±183	219±403	321±639
M_{MPIX}	71±155	80±118	63±145	61±125	163±328	299±635
M_{TRACK}	84±182	85±140	71±141	134±712	51±165	74±218
M_{ADAPT}	20±96	16±37	20±58	22±52	14±26	75±319

Table 1 Evaluation of four reconstruction methods on 17 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. The high error in $S_{s5,r2}$ results from a wide range view, where the sensor produced a high amount of noise.

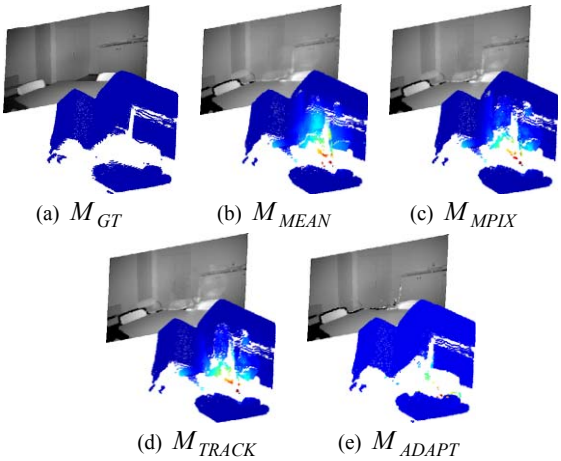


Fig. 8 Results of scene $S_{s2,r1}$ for the evaluated algorithms. In the front the reconstructed 3D static scenes and in the back the accordant 2D images can be seen. (a) Shows the ground truth. In (b) the reconstruction by simple averaging, in (c) the reconstruction by excluding moving pixels, and in (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 proposed method. The colors encode the error of the model if compared to the ground truth – blue means small and red means big error.

In Fig. 8 the resulting static scenes for one example vista space are presented. The figure shows the resulting 3D static scene from the different background modeling techniques and the 2D image created from this model. The colors encode the error of the models compared to the ground truth, where blue denotes a small error and red a big error. The naive background modeling strategies failed in removing the person correctly in all frames, which results in a big error at those positions of the 3D point cloud, where the person is still visible. This gets apparent as a ghost appears at the same positions in the 2D image. The approach presented in this paper reliably removes the person, which provides a sound background model. Table 1 shows an analysis of the arising errors from the background modeling strategies.

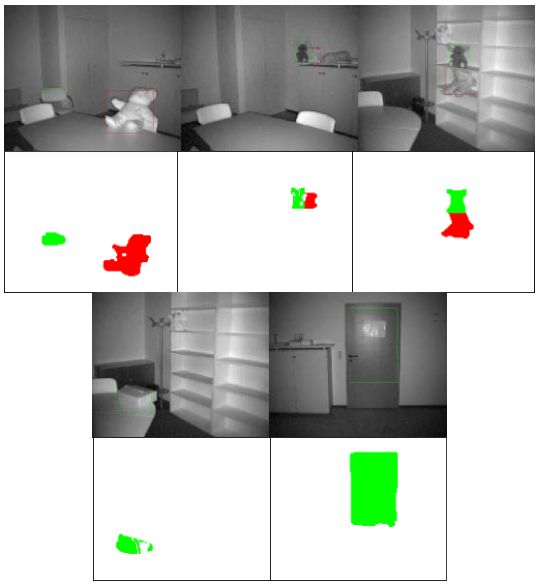


Fig. 9 The images show diverse objects detected by our method. All presented objects have been moved around by the human in the scene. Different colors encode different objects. The pictures show nicely the huge variability in detecting movable objects due to our model independent approach.

The first value is the mean Euclidean distance in mm over all pixels compared to the ground truth and the second value denotes the corresponding standard deviation. The presented errors affirm the viewable impression from Fig. 8 as M_{ADAPT} results in the lowest error rates. The rates are promising with an error mostly at 2cm and never above 10cm. Even in scene $S_{s4,r4}$, 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. A mostly low standard deviation stands for good results in each point as well.

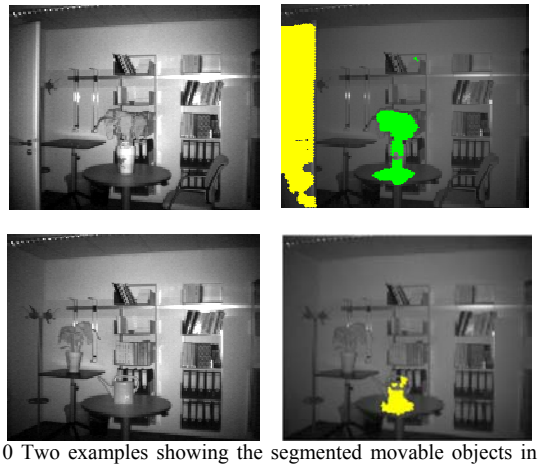
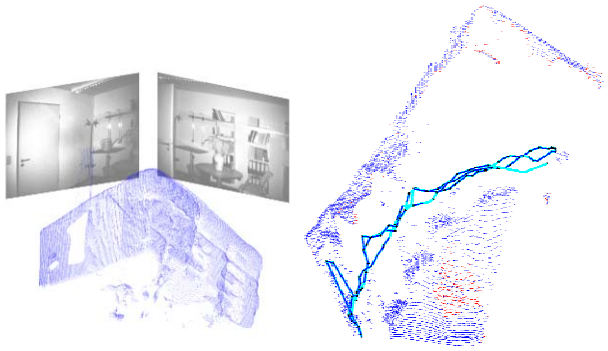


Fig. 10 Two examples showing the segmented movable objects in a 2D image. The first and the third image are the original images and the second and fourth show the marked object. The colored areas belong to different recognized objects, which have been moved at least one time.

Higher error rates ($S_{s5,r2}$) could occur due to noise arising from the 3D sensor, if the observed scene has some disadvantageous characteristics. The sensor has increasing noise per distance and it is sensitive to reflecting and black surface. You can see this in the mentioned figure in the open door in frame 1 and 26.



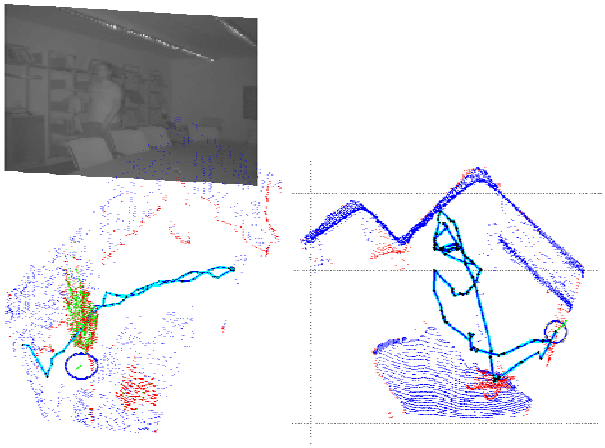
(a) Combined vista spaces (b) Tracks from human movements

Fig. 11 (a) Subsequent vista spaces can be combined by a transformation of the particular spaces in one world coordinate system. The transformation is extracted out of the motion of the robot. (b) All tracks of a human walking behind a table (in cyan). The human walked three times back and forth.

Fig. 9 gives some examples of the detected articulated scene parts. The found objects are color-coded in the image and they are separated from the background to show the variability in detecting diverse objects due to our model independent approach.

The objects can be marked directly in the 2D image (see Fig. 10), which could be used by further processes to calculate more precise information like the shape or texture of the objects.

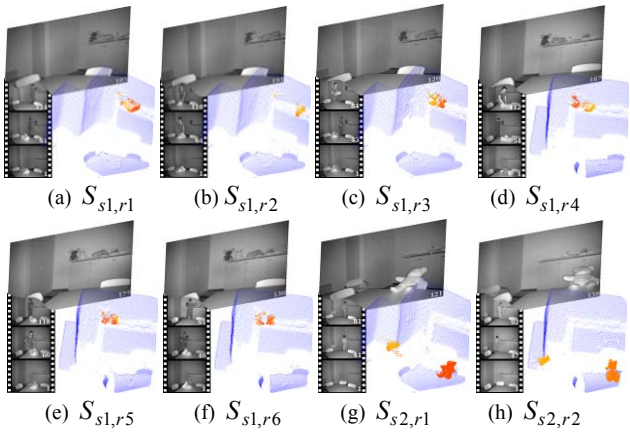
Fig. 11 presents an example of a combination of two subsequent vista spaces. Here, we transform the vista spaces into the same world coordinate system by incorporating the movement of the robot. The two images in the back belong to the different vista spaces. The reliability is dependent on the amount of the movement of the camera. For small movements the error averages at 29 - 86 millimeter[40].



(a) Trajectory (b) Trajectory

Fig. 12 In (a)-(b) tracking results of the proposed system are shown (in cyan). In both views the red pixel denote the dynamic and the blue ones the static parts of the scene. The right scene is taken from $S_{s2,r1}$ (see Fig. 13 (g))

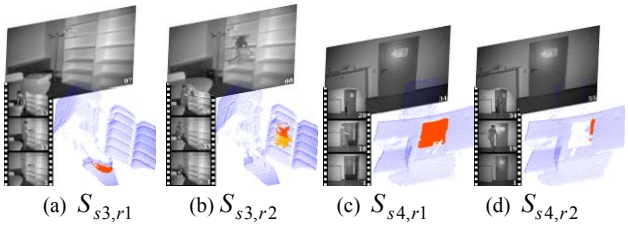
In general using an ICP algorithm with additional loop-closing results in better reconstructions[30,16]. In the second image of Fig. 11 all trajectories from one observation of a vista space are plotted. One human walked three times back and forth. His movements are consistently tracked. Two other example vista spaces and their resulting trajectories are shown in Fig. 12(a)- 12(b) using two different views. The tracking works reliably in most cases. In the remaining cases the inaccuracy can be traced back to the single use of 3D information. The errors result from rapid changes in the movement or specific actions in the scenery. These actions are e. g. found in scene s2, where the person walks to a closet and opens the door. The opening shows a similar point cloud and movement like the person, which stops moving at the same moment. Hence, the hypothesis of the closet is more similar to the person as the person itself and the tracking fails during the opening and closing of the closet.



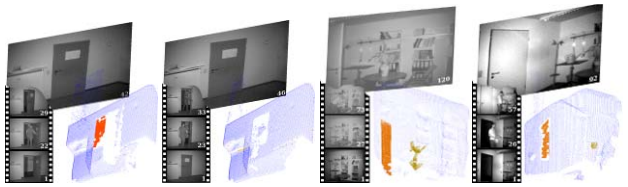
(a) $S_{s1,r1}$ (b) $S_{s1,r2}$ (c) $S_{s1,r3}$ (d) $S_{s1,r4}$

(e) $S_{s1,r5}$ (f) $S_{s1,r6}$ (g) $S_{s2,r1}$ (h) $S_{s2,r2}$

Fig. 13 (a)-(h): 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.



(a) $S_{s3,r1}$ (b) $S_{s3,r2}$ (c) $S_{s4,r1}$ (d) $S_{s4,r2}$



(e) $S_{s4,r3}$ (f) $S_{s4,r4}$ (g) $S_{s5,r1}$ (h) $S_{s5,r2}$

Fig. 14 (a)-(h): 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.

(a) $S_{s6,r1}$ (b) $S_{s7,r1}$

Fig. 15 (a)-(b): 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.

Finally, the articulated scene model for each of the sequences is plotted in Fig. 13-15. In the bottom left a filmstrip gives an idea of the presented sequence, starting at the bottom and ending with the big picture in the background. The corresponding frame numbers are shown in the bottom right in each image. The static background model relates to the blue 3D points and the found articulated parts correspond to the colored areas, whereas different colors encode different objects.

9. Conclusions

We presented in this paper an efficient approach to analyze dynamic scenes directly in 3D. The vista space assumption enables a mobile robot to segment knowledge about the static background, the moving entities and which objects are movable combined in one articulated scene model out of its observations. The gathered knowledge builds a good basement for many following research areas like object learning, navigation or just as an attention on human action spaces. We are going to integrate the static 3D background model in our SLAM approach to realize a better and safer navigation. We are also planning to investigate more work in the detection of the articulations of several objects, like the opening range of a door or the typical movement areas of humans to develop an understanding of safe movement areas or where to pay attention. An example video showing the articulated scene model can be found on the web. *(http://www.techfak.unibielefeld.de/ nbeuter/ArticulatedSceneModel.html)*

Acknowledgement

This work is partly funded by the Cooperative Research Center CRC673 “Alignment in Communication”.

References

1. P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, B. Stankiewicz (2007) Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In *Proceedings of the Symposium on Interaction Challenges for Intelligent Assistants, AAAI Spring Symposium Series. Stanford, CA . AAAI Technical Report SS-07-04*
2. M. Berthold, D. J. Hand (2003) Intelligent Data Analysis, 2nd edn. Springer
3. M.D. Breitenstein, E. Sommerlade, B. Leibe, L. van Gooland I. Reid (2008) Probabilistic parameter selection for learning scene structure from video. In *Proceedings of the British Machine Vision Conference*
4. G. Brostow, I. Essa (1999) Motion based decompositing of video. *IEEE.* **1**:8–13
5. A. Buades, B. Coll, J. M. Morel (2005) A non-local algorithm for image denoising, In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*
6. H. Buxton (2003) Learning and understanding dynamic scene activity, *A review. Image and Vision Computing.* **21**:125–136
7. COGNIRON (2004) The cognitive robot companion, *http://www.cogniron.org.* (FP6-IST-002020)
8. N. Dalal, B. Triggs (2005) Histograms of Oriented Gradients for Human Detection. In *CVPR.* 886–893
9. H. M, Dee, R. Fraile, D. C. Hogg, A. G. Cohn (2008) Modelling scenes using the activity within them. In *Proceedings of the International Conference on Spatial Cognition VI*. 394–408. Springer-Verlag, Berlin, Heidelberg
10. A. Ess, B. Leibe, K. Schindler, L. van Gool (2009) Robust multiperson tracking from a mobile platform, *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(10):1831–1846
11. J. Gibson (1950) The perception of the visual world. *Riverside Press.*
12. L. Guan, J. Franco, M. Pollefeys (2007) 3d occlusion inference from silhouette cues, In, *Computer Vision and Pattern Recognition, CVPR. Ieee, Minneapolis, MN.*
13. L. Guan, M. Pollefeys (2008) A unified approach to calibrate a network of camcorders and tof cameras, 1–12
14. E. Hayman, J. O. Eklundh (2003) Statistical background subtraction for a mobile observer, In, *Proceedings of the International Conference on Computer Vision*. 67–74

15. B. K. Horn, B. G. Schunck (1981) Determining optical flow, In, *Artificial Intelligence.* **17**:185–204
16. B. K. P. Horn, H. Hilden, S. Negahdaripour (1988) Closed-form solution of absolute orientation using orthonormal matrices, *Journal of the optical society America.* **5**(7):1127–1135
17. B. Huhle, P. Jenke, W. Straer (2007) On-the-fly scene acquisition with a handy multisensor-system, In, *Workshop on Dynamic 3D Imaging (Dyn3D)*.
18. B. Huhle, T. Schairer, P. Jenke, W. Straer (2008) Robust non-local denoising of colored depth data, In, *Intl. Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Time of Flight Camera based Computer Vision (TOF-CV)*.
19. K. Kim, T. H Chalidabhongse, D. Harwood, L. Davis (2005) Real-time foreground–background segmentation using codebook model, *Real-Time Imaging.* **11**:172–185
20. J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, R. Klette (2009) Moving object segmentation using optical flow and depth information, In, *Proceedings of the Symposium on Advances in Image and Video Technology.* 611–623
21. K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, T. Darrell (2003) Activity zones for context-aware computing, In, *Proceedings of the International Conference on Ubiquitos Computing, Lecture Notes in Computer Science*. **2864**:90–106
22. B. Kuipers (1999) The spatial semantic hierarchy. Artificial Intelligence, **119**:191–233
23. B. D. Lucas, T. Kanade (1981) An iterative image registration technique with an application to stereo vision, In, *Proceedings of the International Joint Conference on Artificial Intelligence*. 674–679
24. D. Makris, T. Ellis (2003) Automatic learning of an activity-based semantic scene model, In, *Proceedings of the Conference on Advanced Video and Signal Based Surveillance*.
25. S. May, B. Werner, H. Surmann, K. Pervolz (2006) 3D Time-of-Flight cameras for mobile robotics, In, *Intl. Conference on Intelligent Robots and Systems (IROS)*. 790–795
26. A. Mittal, A. Monnet, N. Paragios (2009) Scene modeling and change detection in dynamic scenes: A subspace approach, *Computer Vision and Image Understanding.* **113**(1):63–79
27. D.R. Montello (1993) Scale and multiple psychologies of space, In, *Lecture Notes in Computer Science: Spatial Information Theory A Theoretical Basis for GIS*. **716:**312–321
28. S. Oprisescu, D. Falie, M. Ciuc, V. Buzuloiu (2007) Measurements with tof cameras and their necessary corrections, In, *Intl. Symposium on Signals, Circuits & Systems (ISSCS).*
29. P. Peursum, S. Venkatesh, G. West, H. H. Bui (2004) Using interaction signatures to find and label chairs and floors. Pervasive Computing **3**(4):58–65
30. S. Rusinkiewicz, M. Levoy (2001) Efficient variants of the icp algorithm, In, *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING*.
31. B. C. S. Sanders, T. C. elson, R. Sukthankar (2002) A theory of the quasi-static world, In, *Proceedings of the International Conference on Pattern Recognition*. **3**:1–6
32. I. Schiller, C. Beder, R. Koch (2008) Calibration of a pmd camera using a planar calibration object together with a multi-camera setup, In, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. **37**Part B3a:297–302
33. J. Schmidt, C. Wöhler, L. Krüger, T. Gövert, C. Hermes (2007) 3D scene segmentation and object tracking in multiocular image sequences, In, *Proceedings of the International Conference on Computer Vision Systems*.
34. J. Schmüdderich, V. Willert, J. Eggert, S. Rebhan, C. Goerick, G. Sagerer, E. Körner (2008) Estimating object proper motion using optical flow, kinematics, and depth information, *IEEE Trans Syst Man Cybern B Cybern.* **38**:1139–1151
35. Y. Sheikh, M. Shah (2005) Bayesian modeling of dynamic scenes for object detection, *Transactions on Pattern Analysis and Machine Intelligence.* **27**(11):1778–1792
36. C. Stauffer, W. E. L. Grimson, (1999) Adaptive background mixture models for real-time tracking, In, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 246–252
37. J. Sturm, K. Konelige, C. Stachniss, W. Burgard (2010) Vision-based detection for learning articulation models of cabinet doors and drawers in household environments, In, *Proceedings of the International Conference on Robotics and Automation*.
38. J. Sturm, V. Predeep, C. Stachniss, C. Plagemann, K. Konolige, W. Burgard (2009) Learning kinematic models for articulated objects, In, *Proceedings of the International Joint Conference on Artificial Intelligence*. 1851–1856
39. Swadzba, A., Beuter, N., Schmidt, J., Sagerer, G.: Tracking objects in 6d for reconstructing static scenes. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (2008)
40. A. Swadzba, B. Liu, J. Penne, O. Jesorsky, R. Kompe (2007) A comprehensive system for 3D modeling from range images acquired from a 3d tof sensor, In, *Proceedings of the International Conference on Computer Vision Systems*.
41. A. Swadzba, S. Wachsmuth (2008) Categorizing perceptions of indoor rooms using 3D features, In, *Lecture Notes in Computer Science: Structural, Syntactic, and Statistical Pattern Recognition.* **5342**:744–754 (2008)
42. A. Swadzba, N. Beuter, S. Wachsmuth, F. Kummert (2010) Dynamic 3D Scene Analysis for Acquiring Articulated Scene Models, In, *Proceedings of the International Conference on Robotics and Automation* .
43. X. Wang, K. Tieu, E. Grimson (2006) Learning semantic scene models by trajectory analysis, In, *Proceedings of the European Conference on Computer Vision, lncs*. **3953**:110–123

44. J. Weingarten, G. Gruener, R. Siegwart (2004) A state-of-the-art 3D sensor for robot navigation, In, *Proceedings of the International Conference on Intelligent Robots and Systems*. **3**:2155–2160
45. F. Yuan, A. Swadzba, R. Philippsen, O. Engin, M. Hanheide, S. Wachsmuth (2009) Laser-based navigation enhanced with 3D time-of-flight data, In, *Proceedings of the International Conference on Robotics and Automation*. 2844–2850