2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Vision-based Detection for Learning Articulation Models of Cabinet Doors and Drawers in Household Environments

Jürgen Sturm[1] Kurt Konolige[2] Cyrill Stachniss[1] Wolfram Burgard[1]

Abstract— **Service robots deployed in domestic environments generally need the capability to deal with articulated objects such as doors and drawers in order to fulfill certain mobile manipulation tasks. This however, requires, that the robots are able to perceive the articulation models of such objects. In this paper, we present an approach for detecting, tracking, and learning articulation models for cabinet doors and drawers without using artificial markers. Our approach uses a highly efficient and sampling-based approach to rectangle detection in depth images obtained from a self-developed active stereo system. The robot can use the generative models learned for the articulated objects to estimate their articulation type, their current configuration, and to make predictions about possible configurations not observed before. We present experiments carried out on real data obtained from our active stereo system. The results demonstrate that our technique is able to learn accurate articulation models. We furthermore provide a detailed error analysis based on ground truth data obtained in a motion capturing studio.**

I. Introduction

Home environments are envisioned as one of the key application areas for service robots. Robots operating in such environments often have to deal with articulated objects such as doors or drawers. In the past, several researchers have addressed the problem of estimating and handling doors and drawers [10], [4], [12], [15]. Most of these approaches, however, are either entirely model-free or assume substantial knowledge about the model and its parameters. Whereas model-free approaches release designers from providing any a-priori model information, the knowledge about objects and their articulation properties may greatly support the state estimation and the simulation, planning, and verification of the actions of the robot.

In this paper, we consider the problem of acquiring articulation models of doors and drawers from sequences of depth images acquired with an active stereo camera also presented in this paper. This approach has several advantages. First, it does not rely on artificial markers attached to objects, and second, we do not need to employ expensive range scanners which have have the additional disadvantage that they poorly deal with moving objects, making them inconvenient for learning articulations.

In our concrete scenario, the perception of articulated drawers and doors in a kitchen environment requires the accurate detection of rectangular objects in the depth image sequences, see Fig. 1. We present a highly efficient algorithm that segments the point clouds into planes, and then iteratively fits rectangles to each plane separately. Our perception algorithm can be adapted to the computational capabilities of the robot as it allows to adjust the number of rectangle detections per frame. We furthermore track rectangles over multiple frames. The individual tracks are then used to learn the articulation models. The learning approach instantiates multiple candidate articulation models and selects the one that best explains the data. Once a model has been selected, the robot can use it to predict future configurations of the objects.

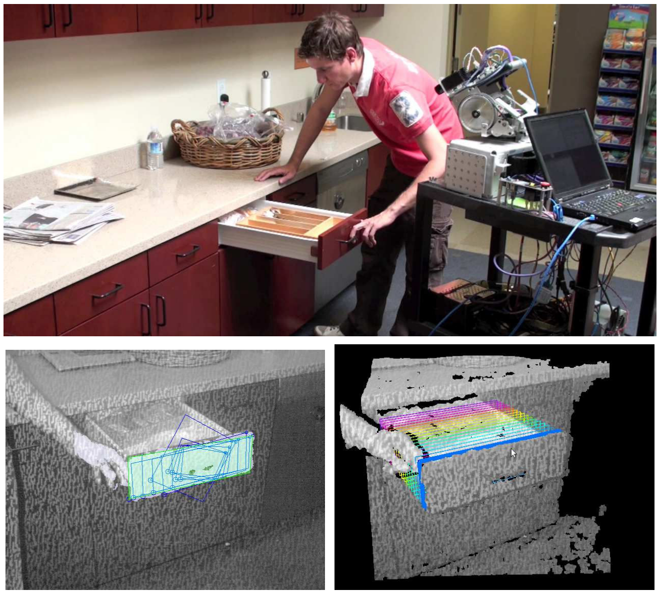

Fig. 1. Top: A drawer is opened and closed and observed with a stereo camera in combination with projected texture. Bottom left: After plane segmentation, we optimize iteratively the pose of a rectangle and evaluate the model fit directly in the disparity image. Bottom right: After combining these detections into a track, we fit an articulation models.

Our implementation has been made available within Willow Garage's open source robotics repository WG-ROS-PKG [23].

This paper is organized as follows. After discussing related work in the following section, we present our camera system, the detection and tracking algorithm as well as the learning approach in Section III. Finally, in Section IV we analyze our approach in different experiments carried out with a real robot in different environments. We furthermore evaluate our method based on ground-truth data obtained with a motion capture system.

[1] Jürgen Sturm, Cyrill Stachniss and Wolfram Burgard are with the Autonomous Intelligent Systems Lab, Computer Science Department, University of Freiburg, Germany. {sturm, stachnis, burgard} @informatik.uni-freiburg.de

[2] Kurt Konolige is with Willow Garage Inc., Menlo Park, CA 94025, USA. konolige@willowgarage.com

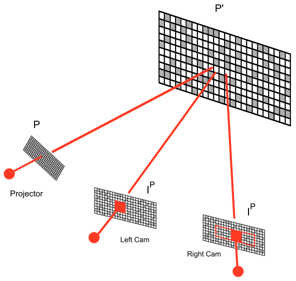

Fig. 2. Our projector and stereo camera system. A pattern P is projected onto a surface to produce P', which is imaged by a left and right camera. To compute depth, the small red block in the left image is matched against a range of blocks in the right image at the same vertical offset, indicated by the outlined rectangle.

II. Related Work

A. Projected Texture Stereo

For our application, we require real-time (15 Hz), accurate and dense point clouds of the scene. Flash ladars [2] often have poor depth and spatial resolution, and have non-Gaussian error characteristics that are difficult to deal with. Line stripe systems [8], [18] have the requisite resolution but cannot achieve 15 Hz operation, nor deal with moving objects. Monocular structured light systems [21] can achieve reasonable frame rates and can sometimes deal with object motion, but still rely on expensive and high-powered projection systems, while being sensitive to ambient illumination and object reflectance.

Stereo systems that employ matching algorithms to produce dense results [6], [14], [27] can be a suitable sensor for our application. However, passive stereo suffers from the problem of *dropouts*: areas of low texture cannot be matched correctly. An interesting and early technology is the use of stereo with structured light [17], [16]. Unlike structured light systems with single cameras, stereo does not depend on the relative geometry of the light pattern – the pattern just lends texture to the scene. Hence the pattern and projector can be simplified, and standard stereo calibration techniques can be used to obtain accurate and dense 3D measurements.

One variant of this technique, known as Spacetime Stereo (STS) [9], [31], varies the pattern over time and integrates several frames. It produces outstanding results on static scenes and under controlled illumination conditions, but moving objects create difficulties [31], [28], [25].

We have developed a compact projector for active stereo with a fixed, random pattern. It provides a texture for stereo that produces excellent error characteristics at distances up to 3 meters, even for surfaces with low reflectivity, see Fig. 2 and Fig. 3.

B. Model-based Detection

Locating objects from 2D images has a long history in computer vision research [5]. Recent approaches for door detection from camera images include [1] and [3]. For 3D point clouds, Hough transforms [26], EM-based algorithms [30] and RANSAC-based approaches [20] have been used successfully for line and plane fitting.

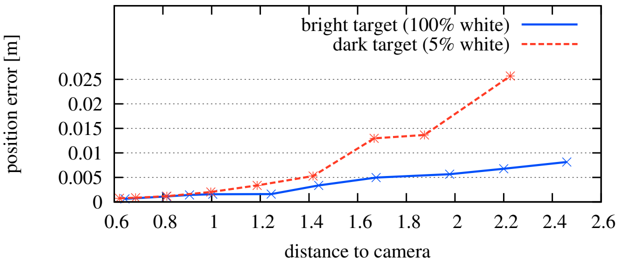

Fig. 3. Positional error of a planar target, observed with our active stereo system. For a white target, the error stays below 2mm until after 1.2m, then goes up to about 1cm at 2.5m. For a very dark target, error is low close up, then becomes larger at distance, when the pattern is difficult to see.

For this work, we evaluated several of the above approaches w.r.t. their applicability to the depth data from our active stereo camera system. We found that RANSAC-based plane segmentation combined with iterative fitting yielded the most robust and accurate results in our context.

C. Learning Articulation Models

Yan and Pollefeys [29] present an approach for learning the structure of an articulated object from feature trajectories under affine projections. Other researchers have addressed the problem of identifying different object parts from image data. Ross *et al.* [19] use multi-body structure from motion to extract links from an image sequence and then fit an articulated model to these links using maximum likelihood learning. There exist several approaches where tracking articulated objects is the key motivation and often an a-priori model is assumed. Comport *et al.* [7], for example, describe a framework for visual tracking of parametric non-rigid multi-body objects based on an a-priori model of the object including a general mechanical link description. Katz *et al.* [13] learn planar kinematic models for various articulated objects in 2D using a KLT tracker. The approach of Schulz *et al.* [22] utilizes prior knowledge about the position and articulation parameters of doors to estimate their state within a Bayesian filtering framework. Kragic *et al.* [15] describe an integrated navigation system for mobile robots which includes a vision-based system for the detection of door handles that enables the robot to successfully open doors. Anguelov *et al.* [4] model doors as line segments that rotate around a hinge. EM is then used to find the model parameters both from 2D range data and images.

In our previous work [24], we learned articulation models for various objects in full 3D using artificial markers. In this paper, we present an extension of our previous approach, that allows us to observe object parts in 3D directly from depth images and thus learn the models without requiring artificial markers. We regard this is as an essential requirement for real-world applications.

III. Approach

In this section, we first briefly describe the structured light approach to obtain dense depth images from stereo. We then present our sampling-based rectangle detector for point

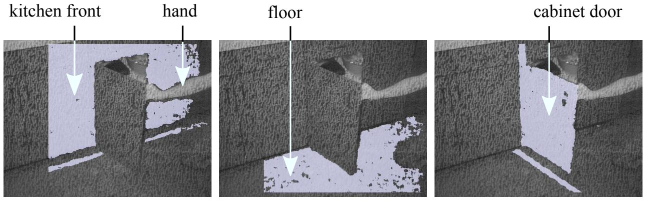

Fig. 4. Finding the tree most prominent planes with our RANSAC-based approach.

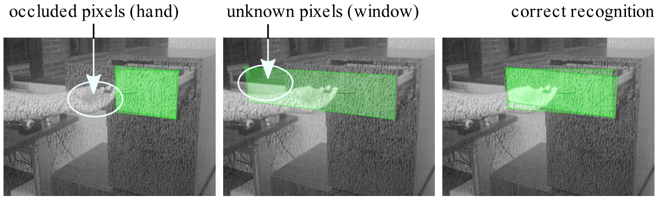

Fig. 5. Effect of the cost parameter for unknown and occluded pixels. Left: cost too high (1.0). Middle: cost too low (0.0). Right: good (0.2).

clouds before we illustrate how the individual observations can be combined into consistent tracks. Finally, we show how articulation models can be learned from such tracks.

A. Dense Depth Images from Stereo and Projected Texture

We consider a projector and a standard calibrated stereo camera configured to be as nearly coincident as possible (see Fig. 2). For simplicity, the focal length of the projector and camera are similar, so that at any distance the projected pattern appears to be the same size in the camera images. A compact, high-power LED device projects a fixed pattern P as a random grid of black and white squares, in sync with the camera exposure. When it is seen by a camera, the pattern produces an image. We use a standard block-matching algorithm to compute the disparity of each pixel [14], that is, the offset between the left and right images. The algorithm runs at 15 Hz for 640x480 at 128 disparities on a single 2 GHz Pentium core.

We tested the device with a 50 deg field of view, using both white and 5% reflectance black planar targets at different distances. The error is taken to be the standard deviation from the best-fit plane. From Fig. 3, the system shows very low error, even out to 2.5 m. For the white target, the error stays below 1 cm throughout this range. Some of the error at the larger distances comes from calibration, as the reconstructed plane will not be perfectly flat. Up to over 1 m, the error is about 2 mm, which is good enough to reconstruct even fine objects. Even with a very dark (5% reflectance) target, the system gives good results up to 2 m, with some increase in error at the larger distance.

An issue with very dark targets is that the block-matching (correlation) response becomes less reliable, as all blocks match equally well. We use an ambiguity test to filter unreliable matches: the ratio between the lowest and 2nd-lowest correlation for a given block in the left image to its candidate correspondents in the right image. Experimentally we determined a threshold for this ratio that excludes most bad matches. For the white target, every pixel made this cutoff. For the dark target, filtered pixels start to occur at 1.2 m, and increase linearly to 2.5 m, when there are no pixels that make the cutoff.

From our stereo processing system, we obtain in each frame a disparity image $D \in \mathbb{R}^{640\times480}$, that contains for each pixel (u, v) its perceived disparity $D(u, v) \in \mathbb{R}$. The relationship between 2D pixels in the disparity image and 3D world points is defined by the projection matrices of the calibrated stereo camera, and can be calculated by a single matrix multiplication from the pixel coordinates and disparity.

B. Model-based Perception from Depth Images

1) Sampling planes: Our RANSAC-based plane fitting algorithm samples three pixels from the depth image, computes from them a plane with coefficients $\mathbf{z}_{\text{plane}} \in \mathbb{R}^4$, and then counts the inliers of that plane. We define the plane to comprise all pixels that are within a certain distance d of the plane, i.e., for which the following holds:

$$\|\mathbf{z}_{\text{plane}}(x\ y\ z\ 1)^T\| \leq d. \tag{1}$$

In general, d depends on the particular noise level of the camera – in our case, we used $d = 0.02$ m. We repeat this process of plane candidate generation until we find a plane with a high enough support, or we exceed a given number of iterations. We select the plane with the most inliers and subtract the corresponding inliers from the point cloud. Subsequently, we apply the same strategy to the remaining points in the cloud, until no more points remain.

For each plane, we create an image mask $M \in \{\text{in-plane}, \text{free}, \text{occluded}, \text{unknown}\}^{640\times480}$ with labels for the pixels in the depth image, i.e.,

$$M(u,v) = \begin{cases} \text{in-plane} & \text{if} \quad \|\mathbf{z}_{\text{plane}}(x\ y\ z\ 1)^T\| & \leq d \\ \text{free} & \text{if} \quad \mathbf{z}_{\text{plane}}(x\ y\ z\ 1)^T & > d \\ \text{occluded} & \text{if} \quad \mathbf{z}_{\text{plane}}(x\ y\ z\ 1)^T & < -d \\ \text{unknown} & \text{otherwise} \end{cases} \tag{2}$$

Here, “in-plane” indicates that the pixel belongs to the plane for which the mask M is computed. In contrast to that, “free” indicates that the observed pixel lies behind the plane and “occluded” that a pixel in front of the plane has been obsersevd which occludes the plane. “Unknown” means that no depth information is available for that pixel.

In contrast to typical approaches to RANSAC-based plane fitting which always assign pixels to one plane, our masks allow points to belong to several planes at the same time. This is useful, as the infinite planes determined via RANSAC always intersect with the subsequent (less significant) planes, thereby cutting out points that make detection of contiguous rectangles more difficult in the next step of the perception process.

For a visualization of the result, see Fig. 4. In this example, our algorithm automatically segmented three planes from a depth image of a cabinet door.

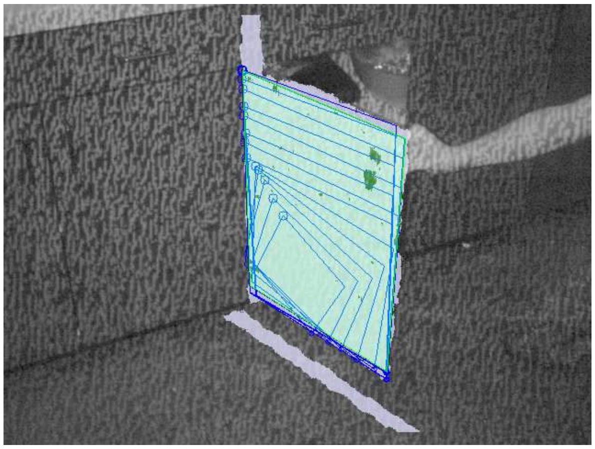

Fig. 6. In each plane, we pick a random starting point from which we optimize iterative the pose and size of the candidate rectangle. In this case, the found rectangle is accepted because both r_{recall} and $r_{\text{precision}}$ are high enough (see text).

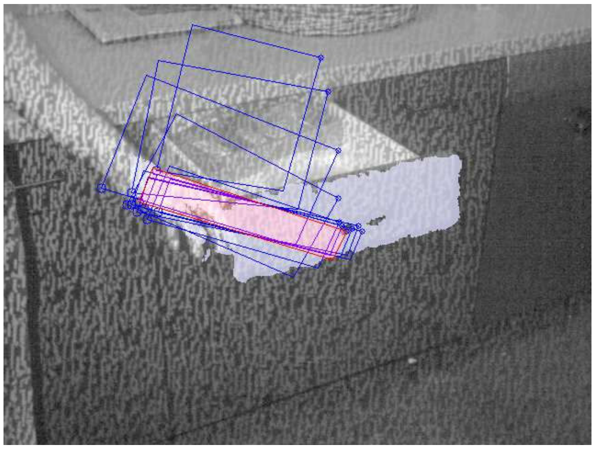

Fig. 7. As our model fitting procedure is greedy, it can get trapped into local maxima. We deal with this problem by starting from multiple starting points. In this case, we reject the found rectangle because r_{recall} is too low.

2) Sampling rectangles: The next step is to find rectangles in the segmented planes. A rectangle in 3D space has 8 degrees a freedom: its position, its orientation and its dimensions (3+3+2). After the plane segmentation, we have already fixed 3 DOFs, so that we need to find the remaining 5 DOFs. We apply an iterative fitting approach here. We start with a sampled candidate rectangle and optimize its pose and size iteratively using an objective function g.

For creating an initial rectangle candidate, we sample a random point from the plane, and sample the other DOFs from a prior distribution. The objective function g is based on the average cost of the pixels inside the rectangle $\mathbf{z}_{\text{rect}} \in \mathbb{R}^8$,

$$g(\mathbf{z}_{\text{rect}}) := -\frac{1}{|\text{pixels}(\mathbf{z}_{\text{rect}})|^{1+\alpha}} \sum_{\text{pixels}(\mathbf{z}_{\text{rect}})} \text{cost}(M(u,v)) \quad (3)$$

The parameter α (that we empirically chose around $\alpha = 0.05$) makes g slightly favor larger rectangles over smaller ones.

Finding a good cost metric cost, in particular for occluded and unknown pixels, is non-trivial. If chosen too low, the greedy search converges on too large rectangles, while a too high cost increases the amount of local maxima in g and in turn leads to the detection of partial rectangles in the presence of occlusions (see Fig. 5).

In each iteration, we now individually optimize every DOF of the rectangle. We apply a small set of discrete changes to each DOF, and evaluate the objective function on $\mathbf{z}'_{\text{rect}}$. If $g(\mathbf{z}'_{\text{rect}}) > g(\mathbf{z}_{\text{rect}})$, we continue with the improved parameter set. When this greedy search converges (or we reach the maximum number of iterations), we need to evaluate the quality of the found match. In preliminary experiments, we found that the value of the objective function was not sufficient for discrimination of false and true positives.

Therefore, we decided to evaluate the rectangle candidate $\mathbf{z}_{\text{rect}}$ using two measures, that are inspired from statistical classification theory and that have a natural interpretation. First, we evaluate the precision $r_{\text{precision}}$ of the rectangle candidate as the ratio of detected pixels and all pixels in the found rectangle. Second, we evaluate the recall r_{recall} as the ratio of pixels in the found rectangle versus the pixels in the selected plane $\mathbf{z}_{\text{plane}}$. For both measures, we used our cost functions to weight occluded and unknown pixels accordingly.

$$r_{\text{precision}}(\mathbf{z}_{\text{rect}}) := \frac{\sum_{\text{pixels}(\mathbf{z}_{\text{rect}})} 1-\text{cost}(M(u,v))}{|\text{pixels}(\mathbf{z}_{\text{rect}})|} \quad (4)$$

$$r_{\text{recall}}(\mathbf{z}_{\text{rect}}) := \frac{\sum_{\text{pixels}(\mathbf{z}_{\text{rect}})} 1-\text{cost}(M(u,v))}{\sum_{\text{pixels}(\mathbf{z}_{\text{plane}})} 1-\text{cost}(M(u,v))} \quad (5)$$

Empirically, we found that a good condition for thresholding is to require that both ratios are above 0.7, which removes most of false positives.

An example of the iterative pose fitting is given in Fig. 6: the rectangle candidate started in the lower left of the door, and iteratively converged to the correct pose and size of the door. The candidate is accepted, because both ratios $r_{\text{precision}}$ and r_{recall} have high values. The greedy search however can get stuck in local maxima. In the example depicted in Fig. 7, the hand is also part of the drawer front plane and the candidate rectangle converged to a rectangle that fits to some extend the hand. Our algorithm then rejects this candidate rectangle because it does not contain the majority of pixels in the plane, i.e., r_{recall} takes a low value.

We deal with the problem of local maxima by starting from several rectangle candidates. In this sense, our algorithm is probabilistically complete, as we would find any visible rectangle in the limit with probability 1. In practice, we chose a fixed number m of samples per plane.

In an early implementation, we approached the problem using a hierarchical model-based approach from computer vision. We looked for edges in the plane mask M of the depth image using the Canny operator, then extracted line segments using the probabilistic Hough transform, and combined neighboring perpendicular line segments to corner candidates. Unfortunately, the projected texture leads to very fringed edges, so that line extraction is unstable; as a consequence, perpendicular line segments are rare, and in many cases no candidates can be created.

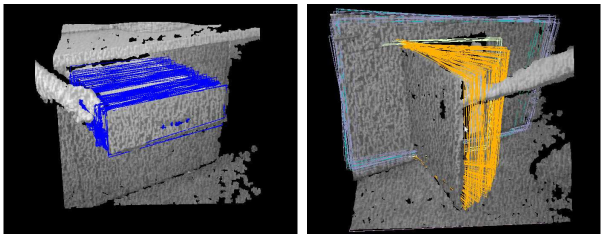

Fig. 8. Observed tracks of a cabinet drawer (left) and a cabinet door (right).

As an alternative to the objective function described in Eq. 3, we evaluated only the contour of the rectangle in the distance transform of the edge image. This technique is called chamfer matching and has been used successfully in computer vision for template-based model matching. In our case, where we have additional uncertainty of the size of the rectangle, we found that the greedy search was very prone to local maxima during the optimization.

C. Tracking

In the remainder of this paper, we drop the subscript in $\mathbf{z}_{\text{rect}} = \mathbf{z}$ to improve readability. The rectangle detector described in the previous section gives us per frame between zero and $n \cdot m$ observations of rectangles (m rectangles in n planes), which need to be integrated into consistent tracks. Checking whether two rectangles $\mathbf{z}_i$ and $\mathbf{z}_j$ are similar requires to take the ambiguity in the representation into account: the same rectangle can be described by eight different parameter vectors (depending on the choice of the corner of origin, and the choice of the front or back side of the rectangle). A track $\mathbf{t}$ is an integrated sequence of k rectangle observations $\mathbf{z}^{1:k} = \mathbf{z}^1, \ldots, \mathbf{z}^k$ that were collected until time t.

In our implementation, we check whether a new observation $\mathbf{z}^{\text{new}}$ (under consideration of the above-mentioned ambiguities) is close to an existing track $\mathbf{t}_i^t$. Then it is either appended to that track $\mathbf{t}_i^{t+1} := \mathbf{z}^1, \ldots, \mathbf{z}^k, \mathbf{z}^{\text{k+1}}$, or a new track is initialized $\mathbf{t}_{\text{new}}^{t+1} := \mathbf{z}^{\text{k+1}}$. For deciding whether a disambiguated observation is close enough to an existing track, we used fixed thresholds on pose change and considered also the uncertainty in the estimate of the object size.

D. Learning Models of Articulated Objects

Our approach for learning models of articulated objects aims at estimating the kinematic nature of the observed tracks of objects in the scene and consists of the following parts:

1) Training Template Models for the Observed Tracks of Object Parts: Since we have no prior information about the nature of the connection between object parts, we do not aim to fit a single model, but instead fit a set of candidate template models representing different kinds of links. This candidate set consists of parameterized models that occur in various objects including a rotational link ($\mathcal{M}^{\text{rotational}}$), a prismatic link ($\mathcal{M}^{\text{prismatic}}$), and a rigid transformation ($\mathcal{M}^{\text{rigid}}$). All models except $\mathcal{M}^{\text{rigid}}$ have a latent variable q that describes the configuration of the link. For a door, the variable q for example describes the opening angle of the door.

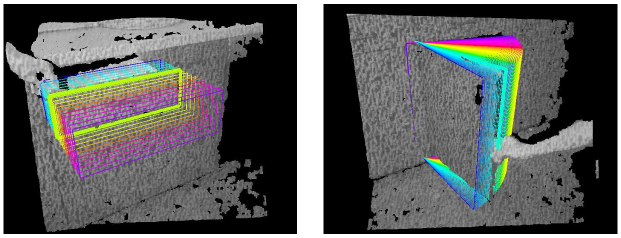

Fig. 9. Left: Articulation model learned from observing a drawer. Right: Same for a door.

Let us consider a track $\mathbf{t}$. To train the candidate models for this track, we have a sequence of k noisy observations $\mathbf{z}^{1:k}$ acquired by the tracker. Each candidate template model has its own training or estimation procedure from the track observations $\mathbf{z}^{1:k}$. For example, for a rotational joint model, we need to estimate the rotation axis and the radius. For further details, we refer the reader to [24].

2) Evaluating a Model: Besides training each model template, we need to evaluate its performance to subsequently select the model that explains the data best.

Let $\mathcal{M}$ be the articulation model describing the observations $\mathcal{D} = \mathbf{z}^{1:k} = \mathbf{z}^1, \ldots, \mathbf{z}^k$ of a track $\mathbf{t}$. To evaluate how well a single observation $\mathbf{z}$ can be explained by a model, we have to determine

$$p(\mathbf{z} \mid \mathcal{M}) = \int_q p(\mathbf{z} \mid q, \mathcal{M}) \; p(q \mid \mathcal{M}) \; dq. \qquad (6)$$

Under the assumption that no latent configuration state q is more likely than another one, this simplifies to

$$p(\mathbf{z} \mid \mathcal{M}) = \int p(\mathbf{z} \mid q, \mathcal{M}) \; dq. \qquad (7)$$

To evaluate $p(\mathbf{z} \mid q, \mathcal{M})$, that is, a measure for how well model $\mathcal{M}$ parameterized by q explains the observation $\mathbf{z}$, we first compute the expected transform

$$\hat{\mathbf{z}} = \mathbb{E}_{\mathcal{M}}[\mathbf{z} \mid q] = f_{\mathcal{M}}(q) \qquad (8)$$

using a model-specific transformation function $f_{\mathcal{M}}(q)$ that computes the expected pose of the object given q. The transformation functions for all template models are described in [24]. Under a Gaussian error assumption, the observation likelihood then becomes

$$p(\mathbf{z} \mid q, \mathcal{M}) \propto \exp\left(-||\hat{\mathbf{z}} - \mathbf{z}||^2/\sigma^2\right) \qquad (9)$$

and finally, we can compute the marginal data likelihood as

$$p(\mathcal{D} \mid \mathcal{M}) = \prod_{\mathbf{z} \in \mathcal{D}} p(\mathbf{z} \mid \mathcal{M}). \qquad (10)$$

3) Model Selection: With the above mentioned approach, we can estimate for each track a set of actuation models $\mathcal{M}^{\text{rigid}}, \mathcal{M}^{\text{prismatic}}, \mathcal{M}^{\text{rotational}}$ and the corresponding observation likelihood using Eq. 10.

For selecting the model, we assign to each learned articulation model a cost that is equal to the negative expected data log-likelihood plus a complexity penalty of the model:

$$\text{cost}_{\mathcal{M}^{\text{type}}} = -\frac{1}{\|\mathcal{D}\|} \log p(\mathcal{D} \mid \mathcal{M}^{\text{type}}) + C(\mathcal{M}^{\text{type}}). \qquad (11)$$

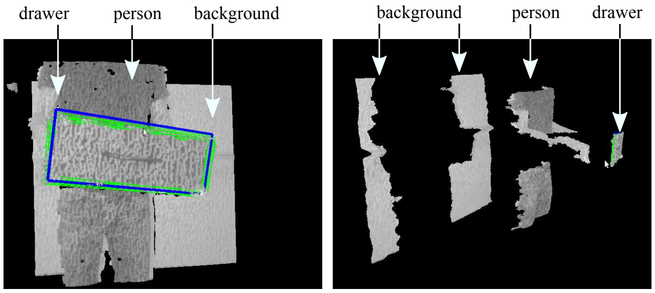

Fig. 10. The blue rectangle shows the ground truth location obtained with a motion capturing studio, while the green rectangles show our estimates.

Then, we select for each track individually the model that has the lowest cost. This articulation model then explains the data of the observed track best while considering at the same time also the model complexity.

IV. Experiments

A. Recognition Capabilities

To evaluate the performance of our sampling-based perception approach, we obtained ground truth pose information from a motion capturing studio. Tracking LEDs were added to an unmounted drawer, and a log file containing 19,412 stereo images including pose information was recorded under a large variety of different poses (see Fig. 10).

As a first result, we found that the drawer was correctly detected in more than 75% of the images up to a distance of 2.3 m from the camera (see Fig. 11 (top)). We also found that the number of significant planes identified via RANSAC that need to be searched increases almost linearly with the distance from the camera. This is an expected result since the drawer appears smaller in the depth image the further it is away.

The average position error of the estimator was on average below 1.5 cm. It also was almost independent of the actual distance to the camera. The same holds for the orientation error, that was on average below 3 deg (see Fig. 11 (bottom)).

In comparison with our previous results [24], the accuracy of our sampling-based perception on active stereo images is approximately five times higher than with the marker-based tracking system [11].

In our current, un-optimized implementation, the plane extraction takes on average 845 ms on a single 2 GHz Pentium core. Creating the image mask of each plane takes approximately 8 ms. Sampling a rectangle candidate from the mask takes 10 ms, optimizing the pose around 313 ms, and finally checking the precision and recall of the candidate consumes another 2.3 ms.

Furthermore, we validated our approach on large number of different doors and drawers in two different kitchens. Also, we successfully tested the detector on a small office pedestal with three drawers of different size, a fuse door and a fire extinguisher door in the wall.

B. Learned Articulation Models

For evaluating the robustness of our articulation model learner, we recorded detailed logfiles of both a door (39.5 cm × 58 cm) and a drawer (39.5 cm × 12.5 cm) of a typical kitchen interior that were repeatedly opened and closed. We recorded a total of 1,023 and 5,202 images. From these logs, we sampled uniformly around 100 images for 50 times, and ran our detector and tracker as described in Sec. III on each of these 50 down-sampled logfiles. For the resulting tracks, we trained the three articulation models and evaluated the outcome of the model selection process (see Fig. 12 (top)).

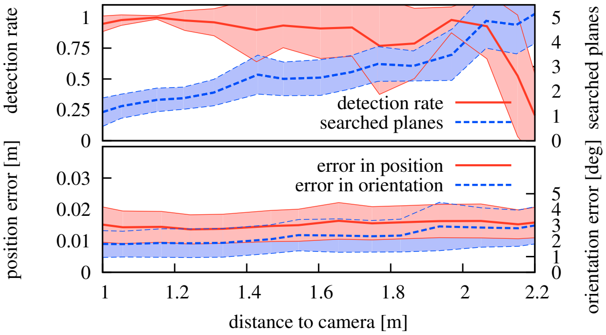

Fig. 11. Evaluation of the detector using ground truth data from the motion capturing studio. Top: Detection rate and number of planes that needed to be searched to find the drawer. Bottom: Accuracy of the pose estimate.

For both datasets, we found that roughly for the first 10 observations, mostly the rigid model was selected, as no substantial motion of the drawer or door was yet detected. The more observations are added to the track, the higher the error between the (rigid) model predictions and the observations becomes. As a result, the prismatic and rotational models are selected more frequently. After 30 observations, model selection has converged in all cases to the true model. For the drawer model we reach a predictive accuracy of 1 cm and 7 deg; for the door we measured a predictive accuracy of 1 cm and 3.5 deg (see Fig. 12 (bottom)). Model fitting and selection takes on average 7 ms, and thus can be easily computed in real-time on a mobile robot.

V. Conclusion

In this paper, we presented an approach for learning articulation models for doors and drawers without requiring artificial markers. Instead, our approach detects and tracks doors and drawers in depth images obtained from a self-developed stereo camera system with structured light. It employs a highly efficient approach to detect and track rectangles in sequences of depth images and uses the resulting tracks to learn accurate articulation models for the corresponding objects. We evaluated our algorithm in extensive experiments also including ground truth data. The results demonstrate that our method is able to achieve high recognition rates and accuracy.

VI. Acknowledgments

The authors gratefully acknowledge the help of everyone at Willow Garage. This work has partly been supported by the European Commission under grant agreement number FP7-248258-First-MM.

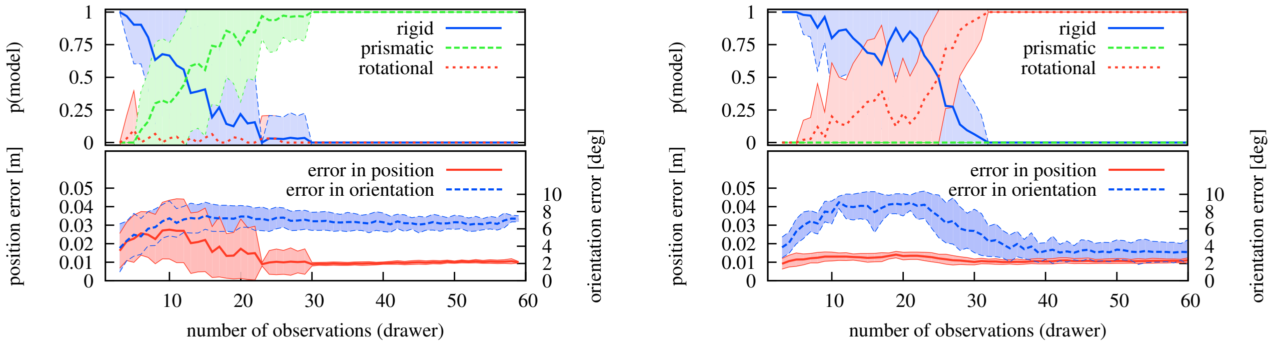

Fig. 12. Evaluation of the articulation models learned for the drawer (left) and the door (right), averaged over 50 runs. The plots at the top show the probability of the articulation model templates, the plots at the bottom show the prediction error of the learned model.

References

[1] J. J. Guerrero A. C. Murillo, J. Kosecka and C. Sagues. Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6):pp. 512–521, 2008.

[2] D. Anderson, H. Herman, and A. Kelly. Experimental characterization of commercial flash ladar devices. In *Proc. of the Int. Conf. of Sensing and Technology*, Palmerston North, New Zealand, 2005.

[3] A. Andreopoulos and J. K. Tsotsos. Active vision for door localization and door opening using playbot. In *Proc. of the Canadian Conf. on Computer and Robot Vision (CRV)*, pages 3–10, Washington, DC, USA, 2008.

[4] D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3777–3784, 2004.

[5] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008.

[6] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined region- and motion-based 3D tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[7] A.I. Comport, E. Marchand, and F. Chaumette. Object-based visual 3D tracking of articulated objects via kinematic sets. In *IEEE Workshop on Articulated and Non-Rigid Motion (CVPRW)*, 2004.

[8] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 1995.

[9] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: a unifying framework dor depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2), February 2005.

[10] A.Y. Ng E. Klingbeil, A. Saxena. Learning to open new doors. In *Proc. of the RSS Workshop on Robot Manipulation*, Seattle, WA, USA, 2009.

[11] M. Fiala. Artag, a fiducial marker system using digital techniques. Technical report, National Research Council Canada, 2004.

[12] A. Jain and C.C. Kemp. Behavior-based door opening with equilibrium point control. In *Proc. of the RSS Workshop on Mobile Manipulation in Human Environments*, Seattle, WA, USA, 2009.

[13] D. Katz, Y. Pyuro, and O. Brock. Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In *Robotics: Science and Systems*, Zurich, Switzerland, 2008.

[14] K. Konolige. Small vision systems: hardware and implementation. In *Proc. of the Int. Symp. on Robotics Research*, pages 111–116, 1997.

[15] D. Kragic, L. Petersson, and H. I. Christensen. Visually guided manipulation tasks. *Robotics and Autonomous Systems*, 40(2-3):193 – 203, 2002.

[16] J. Lim. Optimized projection pattern supplementing stereo systems. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[17] H. K. Nishihara. Prism: A practical real-time imaging stereo matcher. Technical report, MIT, Cambridge, MA, USA, 1984.

[18] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman, and A.Y. Ng. High-accuracy 3d sensing for mobile manipulation: Improving object detection and door opening. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[19] D.A. Ross, D. Tarlow, and R.S. Zemel. Unsupervised learning of skeletons from motion. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2008.

[20] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 2008.

[21] J. Salvi, J. Pages, and J. Batlle. Pattern docification strategies in structured light systems. *Pattern Recognition*, 37(4), 2004.

[22] D. Schulz and W. Burgard. Probabilistic state estimation of dynamic objects with a moving mobile robot. *Robotics and Autonomous Systems*, 34(2-3):107–115, 2001.

[23] J. Sturm. Planar objects package. Willow Garage, Robot Operating System, WG-ROS-PKG, SVN Repository, 2009. Available online at https://code.ros.org/svn/wg-ros-pkg/trunk/sandbox/planar_objects, revision 24017.

[24] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning kinematic models of articulated objects. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2009.

[25] F. Tombari and K. Konolige. A practical stereo system based on regularization and texture projection. In *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO)*, Milan,Italy, 2009.

[26] G. Vosselman and S. Dijkman. 3D building model reconstruction from point clouds and ground plans. *International Archives Photogrammetry and Remote Sensing (IAPRS)*, 34(3W4):37–43, 2001.

[27] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *Proc. of the European Conf. on Computer Vision (ECCV)*, Marseille, France, 2008.

[28] O. Williams, M. Isard, and J. MacCormick. Estimating disparity and occlusions in stereo video sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[29] J. Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[30] Y.L. Yufeng, R. Emery, D. Chakrabarti, and W. Burgard. Using EM to learn 3D models of indoor environments with mobile robots. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, pages 329–336, Williamstown, MA, USA, 2001.

[31] L. Zhang, B. Curless, and S.M. Seitz. Spacetime stereo: shape recovery for dynamic scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.