Pose Estimation of Kinematic Chain Instances via Object Coordinate Regression

Frank Michel
Frank.Michel@tu-dresden.de
Alexander Krull
Alexander.Krull@tu-dresden.de
Eric Brachmann
Eric.Brachmann@tu-dresden.de
Michael Ying Yang
Ying.Yang1@tu-dresden.de
Stefan Gumhold
Stefan.Gumhold@tu-dresden.de
Carsten Rother
Carsten.Rother@tu-dresden.de

TU Dresden
Dresden
Germany

Abstract

In this paper, we address the problem of one shot pose estimation of articulated objects from an RGB-D image. In particular, we consider object instances with the topology of a kinematic chain, i.e. assemblies of rigid parts connected by prismatic or revolute joints. This object type occurs often in daily live, for instance in the form of furniture or electronic devices. Instead of treating each object part separately we are using the relationship between parts of the kinematic chain and propose a new minimal pose sampling approach. This enables us to create a pose hypothesis for a kinematic chain consisting of K parts by sampling K 3D-3D point correspondences. To asses the quality of our method, we gathered a large dataset containing four objects and 7000+ annotated RGB-D frames[1]. On this dataset we achieve considerably better results than a modified state-of-the-art pose estimation system for rigid objects.

1 Introduction

Accurate pose estimation of object instances is a key aspect in many applications, including augmented reality or robotics. For example, a task of a domestic robot could be to fetch an item from an open drawer. The poses of both, the drawer and the item, have to be known by the robot in order to fulfil the task. 6D pose estimation of rigid objects has been addressed with great success in recent years. In large part, this has been due to the advent of consumer-level RGB-D cameras, which provide rich, robust input data. However, the practical use of state-of-the-art pose estimation approaches is limited by the assumption that objects are rigid. In cluttered, domestic environments this assumption does often not hold. Examples are

[1] This dataset will be part of the ICCV 2015 pose challenge: http://cvlab-dresden.de/iccv2015-pose-challenge

Pages 181.1-181.11
DOI: https://dx.doi.org/10.5244/C.29.181

doors, many types of furniture, certain electronic devices and toys. A robot might encounter these items in any state of articulation.

This work considers the task of one-shot pose estimation of articulated object instances from an RGB-D image. In particular, we address objects with the topology of a kinematic chain of any length, i.e. objects are composed of a chain of parts interconnected by joints. We restrict joints to either revolute joints with 1 DOF (degrees of freedom) rotational movement or prismatic joints with 1 DOF translational movement. This topology covers a wide range of common objects (see our dataset for examples). However, our approach can easily be expanded to any topology, and to joints with higher degrees of freedom.

To solve the problem in a straight forward manner one could decompose the object into a set of rigid parts. Then, any state-of-the-art 6D pose estimation algorithm can be applied to each part separately. However, the results might be physically implausible. Parts could be detected in a configuration that is not supported by the connecting joint, or even far apart in the image. It is clear that the articulation constraints provide valuable information for any pose estimation approach. This becomes apparent in the case of self occlusion, which often occurs for articulated objects. If a drawer is closed, then only its front panel is visible. Nevertheless, the associated cupboard poses clear constraints on the 6D pose of the drawer. Similarly, distinctive, salient parts can help to detect ambiguous, unobtrusive parts.

Two strains of research have been prevalent in recent years for the task of pose estimation of rigid objects from RGB-D images. The first strain captures object appearance dependent on viewing direction and scale by a set of templates. Hinterstoisser *et al*. have been particularly successful with LINEMOD [2]. To support articulation, templates can be extracted for each articulation state. In this case, the number of templates multiplies by the number of discrete articulation steps. The multiplying factor applies for each object joint making this approach intractable with a few parts already.

The second strain of research is based on machine learning. Brachmann *et al*. [1] achieve state-of-the-art results by learning local object appearance patch-wise. Then, during test time, an arbitrary image patch can be classified as belonging to the object, and mapped to a 3D point on the object surface, called an object coordinate. Given enough correspondences between coordinates in camera space and object coordinates the object pose can be calculated via the Kabsch algorithm. A RANSAC schema makes the approach robust to classification outliers. The approach was shown to be able to handle textured and texture-less objects in dense clutter. This local approach to pose estimation seems promising since local appearance is largely unaffected by object articulation. However, the Kabsch algorithm cannot account for additional degrees of freedom, and is hence not applicable to articulated objects.

In this work, we combine the local prediction of object coordinates of Brachmann *et al*. with a new RANSAC-based pose optimization schema. Thus, we are capable of estimating the 6D pose of any kinematic chain object together with its articulation parameters. We show how to create a full, articulated pose hypothesis for a chain with K parts from K correspondences between camera space and object space (a minimum of 3 correspondences is required). This gives us a very good initialization for a final refinement using a mixed discriminative-generative scoring function.

To summarize our main **contributions**:
(a) We present a new approach for pose estimation of articulated objects from a single RGB-D image. We support any articulated object with a kinematic chain topology and 1 DOF joints. The approach is able to locate the object without prior segmentation and can handle both textured as well as texture-less objects. To the best of our knowledge there is no competing technique for object instances. We considerably outperform an extension of a

state-of-the-art object pose estimation approach.

(b) We propose a new RANSAC based optimization schema, where K correspondences generate a pose hypothesis for a K-part chain. A minimum of 3 correspondences is always necessary.

(c) We contribute a new dataset consisting of over 7000 frames annotated with articulated poses of different objects, such as cupboards or a laptop. The objects show different grades of articulation ranging from 1 joint to 3 joints. The dataset is also suitable for tracking approaches (although we do not consider tracking in this work).

2 Related Work

In the following, we review four related research areas:

Instance Pose Estimation: Some state-of-the-art instance pose estimation approaches have already been discussed in detail above. The LINEMOD [2] template-based approach has been further improved in the work of Rios-Cabrera and Tuytelaars [10] but the poor scalability in case of articulated objects remains. The approach of Brachmann *et al*. [1] has been combined with a particle filter by Krull *et al*. [4] to achieve a robust tracking system. Although our work fits well in this tracking framework, we consider pose estimation from single images only, in this work. Recently, 6D pose estimation of instances has been executed with a Hough forest framework by Tejani *et al*. [15]. However, in the case of articulated objects the accumulator space becomes increasingly high dimensional. It is unclear whether the Hough voting schema generates robust maxima under these circumstances, or not.

Articulated Instances: Approaches based on articulated iterative closest point [8] can estimate articulated poses given a good initialization, e.g. using tracking. Pauwels *et al*. presented a tracking framework which incorporates a detector to re-initialize parts in case of tracking failure [7]. However, complete re-initialization, e.g. one shot estimation was not shown. Furthermore, the approach relies on key point detectors and will thus fail for textureless objects. Some work in the robotics community has considered the automatic generation of articulated models given an image sequence of an unknown item, e.g. [3, 13]. These approaches rely on active manipulation of the unknown item and observing its behavior, whereas our work considers one-shot pose estimation of an item already known.

Articulated Classes: In recent years, two specific articulated classes have gained considerable attention in the literature: human pose estimation [12, 14] and hand pose estimation [9, 11]. Some of these approaches are based on a discriminative pose initialization, followed by a generative model fit. Most similar to our work is the approach of Taylor *et al*. [14] in which a discriminative prediction of 3D-3D correspondences is combined with a non-linear generative energy minimization. However, the object segmentation is assumed to be given. All class-based approaches are specifically designed for the class at hand, e.g. using a fixed skeleton with class-dependent variability (e.g. joint lengths) and infusing pose priors. We consider specific instances with any kinematic chain topology. Pose priors are not necessary.

Inverse Kinematics: In robotics, the problem of inverse kinematics also considers the determination of articulation parameters of a kinematic chain (usually a robotic arm). However, the problem statement is completely different. Inverse kinematics aims at solving a largely underconstrained system for joint parameters given only the end effector position. In contrast, we estimate the pose of a kinematic chain, given observations of all parts.

3 Method

We will first give a formal introduction of the pose estimation task for rigid bodies and kinematic chains (Sec. 3.1). Then we will continue to describe our method for pose estimation, step by step. Our work is inspired by Brachmann *et al*. [1]. While our general framework is similar, we introduce several novelties in order to deal with articulated objects. The framework consists of the following steps. We use a random forest to jointly make pixel wise predictions: *object probabilities* and *object coordinates*. We will discuss this in Sec. 3.2. We utilize the forest predictions to sample pose hypotheses from 3D-3D correspondences. Here we employ the constraints introduced by the joints of articulated objects to generate pose hypotheses efficiently. We require only K 3D-3D point correspondences for objects consisting of K parts (a minimum of 3 correspondences is required) (Sec. 3.3). Finally, we use our hypotheses as starting points in an energy optimization procedure (Sec. 3.4).

3.1 The Articulated Pose Estimation Task

Before addressing articulated pose estimation, we will briefly reiterate the simpler task of 6D rigid body pose estimation. The objective is to find the rigid body transformation represented by H which maps a point $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^3$ from object coordinate space to a point $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^3$ in camera coordinate space. Transformation H is a homogeneous 4×4 matrix consisting of a rotation around the origin of the object coordinate system and a subsequent translation. In the remainder of this work we assume for notational convenience that the use of homogeneous or inhomogeneous coordinates follows from context.

In the following, we will describe the task of pose estimation for a kinematic chain. A kinematic chain is an assembly of K rigid parts connected by articulated joints. We denote each part with an index $k \in \{1,\ldots,K\}$. We will only consider 1 DOF (prismatic and revolute) joints. A drawer, that can be pulled out of a wardrobe is an example of a prismatic joint. A swinging door is an example of a revolute joint. To estimate the pose of a kinematic chain $\hat{H} = (H_1,\ldots,H_K)$ we need to find the 6D pose H_k for each part k. The problem is however constrained by the joints within the kinematic chain. Therefore, we can find the solution by estimating one of the transformations H_k together with all 1D articulations $\theta_1 \ldots, \theta_{K-1}$, where θ_k is the articulation parameter between part k and $k+1$. The articulation parameter can be the magnitude of translation of a prismatic joint or the angle of rotation of a revolute joint. We assume the type of each joint and its location within the chain to be known. Additionally, we assume the range of possible articulation parameters for all joints to be known. Given θ_k we can derive the rigid body transformation $A_k(\theta_k)$ between the part k and $k+1$. The transformation $A_k(\theta_k)$ determines the pose of part $k+1$ as follows: $H_{k+1} = H_k A_k(\theta_k)^{-1}$. We can use this to estimate the 6D poses of all parts and thus the entire pose $\hat{H}$ of the chain from a single part pose together with the articulation parameters.

3.2 Object Coordinate Regression

As in the work of Brachmann *et al*. [1] we train a random forest to produce two kinds of predictions for each pixel i. Given the input depth image, each tree in the forest predicts object probabilities and object coordinates (both will be discussed later in detail) for each separate object part of our training set.

To produce this output, a pixel is passed trough a series of feature tests which are arranged in a binary tree structure. The outcome of each feature test determines whether the pixel is

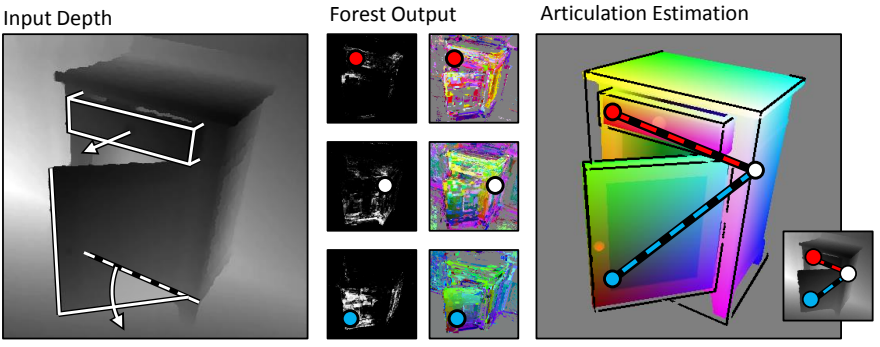

Figure 1: **Articulation estimation.** Left: Input depth image, here shown for the cabinet. The drawer is connected by a prismatic joint and the door is connected by a revolute joint (white lines are for illustration purposes). Middle: Random forest output. Top to bottom: Drawer, base, door, where the left column shows part probabilities and the right the object coordinate predictions, respectively. Right: Articulation estimation between the parts of the kinematic chain using 3D-3D correspondences between the drawer / base and door / base. Note that the three correspondences (red, white, blue) are sufficient to estimate the full 8D pose.

passed to the left or right child node. Eventually, the pixel will arrive at a leaf node where the predictions are stored. The object probabilities stored at the leaf nodes can be seen as a soft segmentation for each object part whereas object coordinate predictions represent the pixel's position in the local coordinate system of the part. Object probabilities from all trees are combined for each pixel using Bayes rule as in [1]. The combined object probabilities for part k and pixel i are denoted by $p_k(i)$.

To generate the object coordinate prediction to be stored at a leaf we apply mean-shift to all samples of a part that arrived at that leaf and store all modes with a minimum size relative to the largest mode. As a result we obtain multiple object coordinate predictions $\mathbf{y}_k(i) = (\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k)^\top$ for each tree, object part k and pixel i. The terms x_k, y_k, and z_k shall denote the coordinates in the local coordinate system of part k. We adhere exactly to the training procedure of [1] but choose to restrict ourselves to depth difference features for robustness.

3.3 Hypothesis Generation

We now discuss our new RANSAC hypotheses generation schema using the forest predictions assuming that $K = 3$. We will consider kinematic chains with $K = 2$ or $K > 3$ at the end of this section. An illustration of the process can be found in Fig. 1. We draw a single pixel i_1 from the inner part ($k = 2$) randomly using a weight proportional to the object probabilities $p_k(i)$. We pick an object coordinate prediction $\mathbf{y}_k(i_1)$ from a randomly selected tree t. Together with the camera coordinate $\mathbf{x}(i_1)$ at the pixel this yields a 3D - 3D correspondence $(\mathbf{x}(i_1), \mathbf{y}_k(i_1))$. Two more correspondences $(\mathbf{x}(i_2), \mathbf{y}_{k+1}(i_2))$ and $(\mathbf{x}(i_3), \mathbf{y}_{k-1}(i_3))$ are sampled in a square window around i_1 from the neighbouring kinematic chain parts $k+1$ and $k-1$. We can now use these correspondences to estimate the two articulation parameters

θ_{k-1} and θ_k between part k and its neighbours.

Estimating Articulation Parameters. We will now discuss how to estimate the articulation parameter θ_k from the two correspondences $(\mathbf{x}(i_1), \mathbf{y}_k(i_1))$ and $(\mathbf{x}(i_2), \mathbf{y}_{k+1}(i_2))$. Estimation of θ_{k-1} can be done in a similar fashion. The articulation parameter θ_k has to fulfil

$$\|\mathbf{x}(i_1) - \mathbf{x}(i_2)\|^2 = \|\mathbf{y}_k(i_1) - A_k(\theta_k)\mathbf{y}_{k+1}(i_2)\|^2, \tag{1}$$

meaning the squared Euclidean distance between the two points $\mathbf{x}(i_1)$ and $\mathbf{x}(i_2)$ in camera space has to be equal to the squared Euclidean distance of the points in object coordinate space of part k. Two solutions can be calculated in closed form. A derivation can be found in the supplemental note. In case of a revolute joint with a rotation around the x-axes the solutions are:

$$\theta_k^1 = \operatorname{asin}\left(\frac{d_\mathbf{x} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2+b^2}}\right) - \operatorname{atan2}(b,a) \quad \text{and}$$

$$\theta_k^2 = \pi - \operatorname{asin}\left(\frac{d_\mathbf{x} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2+b^2}}\right) - \operatorname{atan2}(b,a). \tag{2}$$

where $d_\mathbf{x} = \|\mathbf{x}(i_1) - \mathbf{x}(i_2)\|^2$ shall abbreviate the squared distance between the two points in camera space. Furthermore $a = 2(\mathsf{y}_k\mathsf{z}_{k+1} - \mathsf{z}_k\mathsf{y}_{k+1})$ and $b = -2(\mathsf{y}_k\mathsf{y}_{k+1} + \mathsf{z}_k\mathsf{z}_{k+1})$. It should be noted that, depending on the sampled point correspondences, θ_k^1 and θ_k^2 might not exist in $\mathbb{R}$ and are thus no valid solutions. Otherwise, we check whether they lie within the allowed range for the particular joint. If both solutions are valid we select one randomly. If no solution is valid, the point correspondence must be incorrect and sampling has to be repeated.

In case of a prismatic joint with a translation along the x-axis we can also solve Eq. (1) in closed form:

$$\theta_k^1 = -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q} \quad \text{and} \quad \theta_k^2 = -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}, \tag{3}$$

where $p = 2(\mathsf{x}_{k+1} - \mathsf{x}_k)$ and $q = (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 - d_\mathbf{x}$. Solutions for prismatic joints with translations along other axes can be found analogously. We check again whether θ_k^1 and θ_k^2 are valid solutions in the allowed range of parameters in $\mathbb{R}$ and repeat sampling if necessary.

Pose Estimation. Once we estimated θ_k and θ_{k+1} we derive $A_k(\theta_k)$ and $A_{k+1}(\theta_{k+1})$ and map the two sampled points $\mathbf{y}_{k+1}(i_2)$ and $\mathbf{y}_{k-1}(i_3)$ to the local coordinate system of part k. We have now three correspondences between the camera system and the local coordinate system of part k, allowing us to calculate the 6D pose H_k using the Kabsch algorithm. The 6D pose H_k together with the articulation parameters yields the pose $\hat{H}$ of the chain.

In case of a kinematic chain consisting of $n > 3$ parts, we start by randomly selecting an inner part k. We recover the 6D pose using the two neighbouring parts as described above. Then, we calculate the missing articulation parameters one by one by sampling one correspondence for each part remaining. In case of a kinematic chain consisting of $n = 2$ parts, we draw a single sample from one part and two samples from the other part.

3.4 Energy Optimization

We rank our pose hypotheses with the same energy function as in [1]:

$$\hat{E}(\hat{H}) = \lambda^{depth}E^{depth}(\hat{H}) + \lambda^{coord}E^{coord}(\hat{H}) + \lambda^{obj}E^{obj}(\hat{H}). \tag{4}$$

The kinematic chain is rendered under the pose $\hat{H}$ and the resulting synthetic images are compared to the observed depth values (for E^{depth}) and the predicted object coordinates (for E^{coord}). Furthermore E^{obj} punishes pixels within the ideal segmentation mask if they are unlikely to belong to the object. Weights λ^{depth}, λ^{coord} and λ^{obj} are associated with each energy term. The best hypotheses are utilized as starting points for a local optimization procedure. Instead of the refinement scheme proposed by [1] we used the Nelder-Mead simplex algorithm [6] within a general purpose optimization where we refine the 6D pose H_k of part k together with all 1D articulations $\theta_1 \ldots, \theta_{k-1}$ of the kinematic chain. We consider the pose with the lowest energy as our final estimate.

4 Experiments

To the best of our knowledge there is no RGB-D dataset which fits our setup, i.e. instances of kinematic chains with 1 DOF joints. Therefore, we recorded and annotated our own dataset and will make it publicly available.

4.1 Dataset

We created a dataset of four different kind of kinematic chains which differ in the number and type of joints. The objects are a laptop with a hinged lid (one revolute joint), a cabinet with a door and drawer (one revolute and one prismatic joint), a cupboard with one movable drawer (one prismatic joint) and a toy train consisting of four parts (four revolute joints).

Test Data. We recorded two RGB-D sequences per kinematic chain with Kinect, resulting in eight sequences with a total of 7047 frames. The articulation parameters are fixed within one sequence but changes between sequences. The camera moved freely around the object, with object parts sometimes being partly outside the image. In some sequences parts were occluded.

Depth maps produced by Kinect contain missing measurements, especially at depth edges and for certain materials. This is a problem in case of the laptop, because there are no measurements for the display which is a large portion of the lid. To circumvent this, we use an off-the-shelf hole filling algorithm by Liu *et al*. [5] to pre-process all test images.

We modelled all four kinematic chains with a 3D modelling tool and divided each object into individual parts according to the articulation. Ground truth annotation for the parts was produced manually, including articulation, for all test sequences. We manually registered the models of the kinematic chains onto the first frame of each sequence. Based on this initial pose an ICP algorithm was used to annotate the consecutive frames, always keeping the configuration of joints fixed. We manually re-initialized if ICP failed.

Training Data. Similar to the setup in [2], we render our 3D models to create training sets with a good coverage of all possible viewing angles. Hinterstoisser *et al*. [2] used a regular icosahedron-based sampling of the upper view hemisphere. Different levels of in-plane rotation were added to each view. Since our training images always contain all parts of the kinematic chain, more degrees of freedom have to be taken into account, and each view has to be rendered with multiple states of articulation. Therefore, we follow a different approach in sampling azimuth, elevation, in-plane rotation and articulation to create images. Since naive uniform sampling could result in an unbalanced coverage of views we chose to deploy stratified sampling. For all kinematic chains we subdivide azimuth in 14, elevation in 7 and the in-plane rotation in 6 subgroups. The articulation subgroups where chosen as

follows: Laptop: 4, Cabinet: 3 (door), 2 (drawer), Cupboard: 4, Toy train: 2 for each joint. For example this results in $14 \times 7 \times 6 \times 4 = 2352$ training images for the laptop.

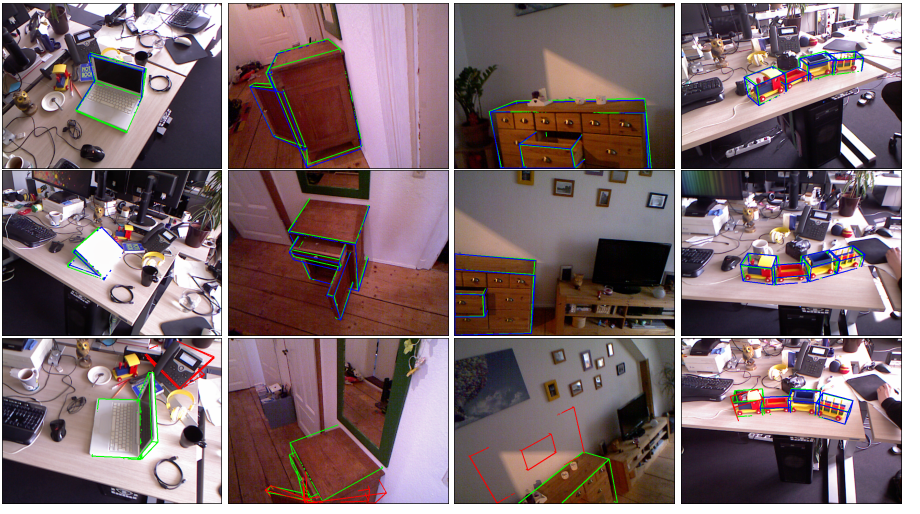

Figure 2: **Our dataset.** These images show results on our dataset. The estimated poses are depicted as the blue bounding volume, the ground truth is shown as the green bounding volume of the object parts. The last row contains cases of failure where the bounding boxes of the estimated poses are shown in red.

4.2 Setup

In this section, we describe our experimental setup. We introduce our baseline and state training and test parameters.

Baseline. We compare to the 6D pose estimation pipeline of Brachmann *et al*. [1]. We treat each object part as an independent rigid object and estimate its 6D pose. This drops any articulation or even connection constrains.

Training Parameters. We use the same parameters as Brachmann *et al*. [1] for the random forest. However, we disabled RGB features because we expect our rendered training set to be not realistic in this regard. On the other hand, to counteract a loss in expressiveness and to account for varying object part sizes, we changed one maximum offset of depth difference features to 100 pixel meters while keeping the other at 20 pixel meters. For robustness, we apply Gaussian noise with small standard deviation to feature responses. In tree leafs we store all modes with a minimum size of 50% with respect to largest mode in that leaf. Mode size means the number of samples that converged to that mode during mean-shift. We train one random forest for all four kinematic chains jointly (11 individual object parts). As negative class we use the background dataset published by Brachmann *et al*. [1]. As mentioned above, training images contain all parts of the associated kinematic chain. Additionally, we render a supporting plane beneath the kinematic chain. Features may access depth appearance of the other parts and the plane. Therefore, the forest is able to learn contextual information. If a feature accesses a pixel which belongs neither to plane nor to a kinematic chain part, random noise is returned. We use the same random forest for our method and the baseline.

Test Parameters. For the baseline we use the fast settings for energy minimization as proposed by [1]: They sample 42 hypotheses and refine the 3 best with a maximum of 20 iterations. We do this for each part of a kinematic chain separately. In contrast, our method does not treat parts separately, but hypotheses are drawn for each kinematic chain in its entirety. Therefore, in our method, we multiply the number of hypotheses with the number of object parts (e.g. $2 \times 42 = 84$ for the laptop). Similarly, we multiply the number of best hypotheses refined with the number of parts (e.g. $2 \times 3 = 6$ for the laptop). We stop refinement after 150 iterations.

Metric. The poses of all parts of the kinematic chain have to be estimated accurately in order to be accepted as a correct pose. We deploy the following pose tolerance [1, 2, 4] on each of the individual object parts k : $\frac{1}{|\mathcal{M}_k|} \sum_{\mathbf{x} \in \mathcal{M}_k} ||H_k\mathbf{x} - \tilde{H}_k\mathbf{x}|| < \tau, k \in \mathcal{K}$, where $\mathbf{x}$ is a vertex from the set of all vertices of the object model[2] $\mathcal{M}_k$, $\tilde{H}_k$ denotes the estimated 6D transformation and H_k denotes the ground truth transformation. Threshold τ is set to 10% of the object part diameter. We also show numbers for the performance of individual object parts. The results are shown in Table 4.2 and discussed below.

Object	Sequence		Brachmann *et al.* [1]	Ours
Laptop	1	all	8.9%	64.8%
		parts	29.8% 25.1%	65.5% 66.9%
	2	all	1%	65.7%
		parts	1.1% 63.9%	66.3% 66.6%
Cabinet	3	all	0.5%	95.8%
		parts	86% 46.7% 2.6%	98.2% 97.2% 96.1%
	4	all	49.8%	98.3%
		parts	76.8% 85% 74%	98.3% 98.7% 98.7%
Cupboard	5	all	90%	95.8%
		parts	91.5% 94.3%	95.9% 95.8%
	6	all	71.1%	99.2%
		parts	76.1% 81.4%	99.9% 99.2%
Toy train	7	all	7.8%	98.1%
		parts	90.1% 17.8% 81.1% 52.5%	99.2% 99.9% 99.9% 99.1%
	8	all	5.7%	94.3%
		parts	74.8% 20.3% 78.2% 51.2%	100% 100% 97% 94.3%

Table 1: Comparison of Brachmann *et al.* [1] and our approach on the four kinematic chains. Accuracy is given for the kinematic chain (all) as well as for the individual parts (parts).

4.3 Results

The baseline can detect individual parts fairly well in case occlusion caused by other parts of the kinematic chain is low to moderate. An example is the performance for both cupboard sequences (Sequences 5 & 6) as well as the individual performance of the first (locomotive) and the third part of the toy train (Sequences 7 & 8). However, the method is not able to handle strong self occlusion. This can be seen in the poor performance of the last part of the toy train (Sequences 7 & 8) and in the complete failure to estimate the pose of the

[2] The vertices of our models are virtually uniform distributed.

cabinet drawer when it is only slightly pulled out (Sequence 3), see Fig. 2 (first row, second column). Providing contextual information between object parts during forest training seems not to be sufficient to resolve self occlusion. Flat objects do not stand out of the supporting plane, which results in noisy forest output. This may explain the rather poor performance of the second part of the toy train which is almost completely visible within the entire test sequences (Sequences 7 & 8).

Our method shows superior results (89% averaged over all sequences and objects) in comparison to the baseline (29%). Employing articulation constraints within the kinematic chain results in better performance on the individual parts as well as for the kinematic chains in its entirety, see Table 4.2. Our approach of pose sampling for kinematic chains does not only need less correspondences, it is also robust when dealing with heavy self occlusion. Even in cases where one part is occluded more than 75%, e.g. the laptop keyboard in Sequence 2, we are still able to correctly estimate the pose of the occluded part, see Fig. 2 (second row, first column). Our approach enables parts with a high quality forest prediction to boost neighbouring parts with a noisy forest prediction (e.g. the second part of the toy train in Sequences 7 & 8).

We compare our approach to the method of [1] in regard of the error of the articulation parameter. Fig. 3 shows results for the cabinet in sequence 4. Poses estimated with our method result in a low error for both the prismatic (translational) as well as the revolute (rotational) joint. As a result the distribution for our approach is peaked closely around the true articulation parameter. This is not the case for the approach of [1]. The peak for the rotational error lies at 3° and the peak for the translation lies at +5mm.

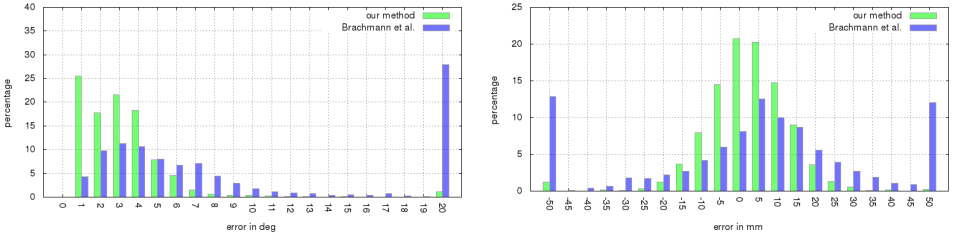

Figure 3: Histogram of rotational and translational error of our approach compared to [1] for the cabinet (sequence 4)

5 Conclusion

We presented a method for pose estimation of kinematic chain instances from RGB-D images. We employed the constraints introduced by the joints of the kinematic chain to generate pose hypotheses using K 3D-3D correspondences for kinematic chains consisting of K parts. Our approach shows superior results when compared to an extension of state-of-the-art object pose estimation on our new dataset. This dataset is publicly available under *http://cvlab-dresden.de/research/scene-understanding/pose-estimation/#BMVC15*. The proposed method is not restricted to a chain topology. Therefore, in future work, we will address the extension to arbitrary topologies and joints with higher degrees of freedom.

Acknowledgements. We thank Daniel Schemala, Stephan Ihrke, Andreas Peetz and Benjamin Riedel for their help preparing datasets and their contributions to our implementation.

References

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[2] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV*, 2012.

[3] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *ICRA*, 2013.

[4] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-dof model based tracking via object coordinate regression. In *ACCV*, 2014.

[5] J. Liu, X. Gong, and J. Liu. Guided inpainting and filtering for kinect depth maps. In *ICPR*, 2012.

[6] J. A. Nelder and R. Mead. A simplex method for function minimization. In *Computer Journal*, 1965.

[7] K. Pauwels, L. Rubio, and E. Ros. Real-time model-based articulated object pose detection and tracking with variable rigidity constraints. In *CVPR*, 2014.

[8] S. Pellegrini, K. Schindler, and D. Nardi. A generalisation of the icp algorithm for articulated bodies. In *BMVC*, 2008.

[9] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. 2014.

[10] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3D object detection: A real time scalable approach. In *ICCV*, 2013.

[11] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. CHI, 2015.

[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011.

[13] J. Sturm, A. Jain, C. Stachniss, C. C. Kemp, and W. Burgard. Operating articulated objects based on experience. In *IROS*, 2010.

[14] J. Taylor, J. Shotton, T. Sharp, and A.W. Fitzgibbon. The Vitruvian Manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*, 2012.

[15] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *ECCV*, 2014.