2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Active Scene Recognition for Programming by Demonstration using Next-Best-View Estimates from Hierarchical Implicit Shape Models

Pascal Meißner, Reno Reckling, Valerij Wittenbeck, Sven R. Schmidt-Rohr and Rüdiger Dillmann

Abstract— **We present an approach that combines passive scene understanding with object search in order to recognize scenes in indoor environments that cannot be perceived from a single point of view. Passive scene recognition is performed using Implicit Shape Models based on spatial relations between objects. ISMs, a variant of the Generalized Hough Transform, are extended to describe scenes as sets of objects with relations lying between them. Relations are expressed as six-degree-of-freedom (DoF) relative object poses. They are extracted from sensor recordings of human demonstrations of actions usually taking place in the corresponding scene. In a scene ISMs solely represent relations of n objects towards a common reference. Violations of other relations are not detectable. To overcome this limitation, we extend our scene model, using hierarchical agglomerative clustering, to a binary tree consisting of ISMs. Active scene recognition aims to simultaneously detect present scenes and look for objects these scenes consist of. For a pivoting stereo camera rig, we achieve this by performing recognition with ISMs in an object search loop using next-best-view (NBV) estimates. A criterion, on which we greedily choose views the rig shall adopt next, is the confidence to detect objects in them. In each step during the search, confidences on potential positions of objects, not found yet, are calculated based on the best available scene hypothesis. This is done by reversing the principle of ISMs and using spatial relations to predict potential object positions starting from the objects already detected.**

I. INTRODUCTION

Programming by Demonstration (PbD) is a robot learning approach that aims to acquire knowledge about everyday actions from demonstrations performed by humans and recorded by sensors. For a robot to employ this knowledge autonomously, it has to be able to adapt it to various contexts. From this fact follows that robots need capabilities to assess their environment. A crucial aspect are the geometrical scenes in which a robot operates. Characteristics of indoor environments are best described by the objects they contain, as Quattoni et al. [1] argue for two-dimensional images. In manipulation scenarios, not only object occurrences, but also spatial relations, being described by relative object poses in six DoF, are indispensable to characterize scenes.

Due to their spatial extent and present clutter, in general we cannot assume that scenes are perceivable from a single point of view. Therefore we present an approach that recognizes scenes based on detected objects and searches further objects, not found yet, to improve the scene estimation gotten beforehand. The large search space of viewpoints and sensor orientations, coming into question to detect objects in scenes, as well as travel costs coming with actions and object detector runtimes, make uninformed search infeasible in this setting. Models of scenes, based on spatial relations, provide a mean to restrict that search space. Object search, guided by minimizing costs and maximizing expectation of detecting objects, may choose actions looking multiple search steps ahead. We preferred a next-best-view approach looking one step ahead following arguments in [2]. It states that in object search, frequent measurements affect available search policies so much, that constant replanning is necessary.

In the following, we present an active scene recognition system that deals with the issues, discussed so far. In Sec. III to V, we describe a passive scene recognition approach that assumes input data is given. In Sec. VI to VII we introduce an search approach giving up this assumption and performing the presented passive scene recognition in a loop. Therefore passive scene recognition needs to run in real time. We learn our scene models following the principle of PbD, but we differ from learning manipulations in that we only record object configurations before and after actions take place and not in-between. Therefore occlusions during object manipulation have no impact on our learning methodology.

II. RELATED WORK

Scene understanding, a computer vision problem, is tackled in two ways: Algorithms either rely on results from object detection or they infer scene information directly from image data without referring to intermediate concepts. Methods corresponding to the latter paradigm employ global image descriptors as the gist descriptor [3] or use local descriptors, calculated on specific image regions [4] that are determined beforehand. Such approaches work well for outdoor scenes, but deliver poor performance on indoor scenes.

Scene understanding, based on objects, is more concerned with modeling correlations between identities of objects, than capturing details of spatial relations between them. One of the few relation-oriented approaches is [5] that maps relative object poses on symbolic qualitative relations, defined by hand. Symbolic relations generalize well, but lose details in the topologies of object relations as needed in our application area. In contrast, numerous methods deal with learning spatial relations in the field of part-based object recognition. Supervised learning of object configurations is performed with Constellation Models in [6], where positions are represented using normal distributions, ignoring object orientations. As in [7], where unsupervised learning of scenes takes place using Bayesian non-parametric models, parametric modeling limits expressiveness concerning spatial relations. A method that is able to represent the complex relations encountered in our

P. Meißner, R. Reckling, V. Wittenbeck, S. R. Schmidt-Rohr and R. Dillmann are with Institute of Anthropomatics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. pascal.meissner@kit.edu

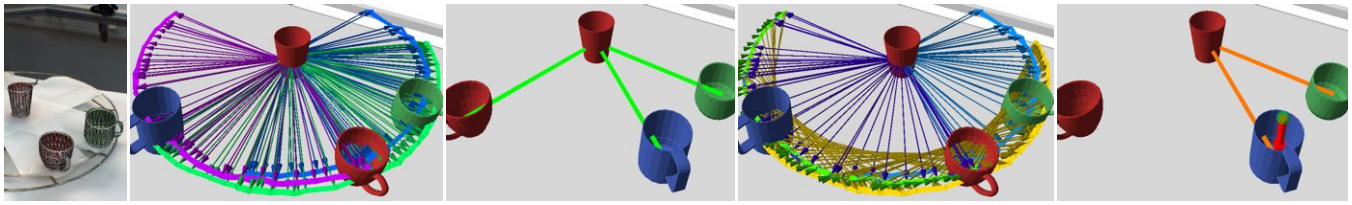

Fig. 1. Cups rotate around a static object in fixed relations towards each other. Center-left: Recorded trajectories & relative object poses towards center, resulting from learnt non-hier. ISM (arrows). Center: ISM fully detects scene despite relation between angled red & blue cup violated. Center-Right: Heuristic (Sec. V-A) detects relation between red & blue cup (yellow arrows). Others not due to high noise. Right: Relation violation found by hier. ISM.

scenario, are Implicit Shape Models (ISMs) [8]. Being non-parametric, they model relative object positions in any type of complexity. Orientation information is neglected.

Another field in computer vision is view planning which deals with optimizing information gain in capturing image resp. range data by autonomously adapting sensor parameters. Most approaches in this area focus on choosing view points and only look one step ahead instead of considering entire sequences of actions. These assumptions lead to a research topic known as next-best-view problem. Reconstructing object surfaces is a popular application for next-best-view optimization. In [9], view points for a moving laser scanner are chosen based on already measured data in order to optimize information gain in terms of sensing unknown space. On a larger scale, [10] deals with the similar task of scanning an initially unknown environment with a mobile robot. Both approaches derive actions without considering prior knowledge like scene models.

A less common task in view planning is three-dimensional object search. In [11], objects are being looked for based on prior knowledge. Both detection likelihoods and action costs are taken into account to choose next-best navigation and camera orientation actions. Instead of searching configurations of objects as in our application, it is limited to searching a single object. [2] deals with a similar problem, but considers action sequences as well as uncertainties in decision-making. A Partially Observable Markov Decision Process (POMDP) is used to handle that additional complexity in order to enable a robot to find objects in cluttered scenarios. Object locations are modeled as six DoF poses, but in a parametric manner and based on absolute poses instead of spatial relations. To be able to cope with the high runtimes of a POMDP, important simplifications in the planning model, e.g. regarding the size of the action space, are made.

III. SCENE, OBJECTS AND DATA ACQUISITION

We define a scene $\mathbf{S} = (\{o\}, \{\mathbf{R}\})$ as a set of spatial relations $\{\mathbf{R}\}$ between pairs of objects o_j and o_k, including the objects as well. Relations $\mathbf{R}$ consist of sets of relative object poses $\{\mathbf{T}_{jk}\}$ that are represented as transformation matrices $\mathbf{T} \in \mathbb{R}^{4x4}$. Matrices $\mathbf{T}$ capture all six DoF in space and include positions $\mathbf{p} \in \mathbb{R}^3$. Models of scenes are learnt from demonstrations during which trajectories $\mathbf{J}(o) = (\mathbf{E}(o,1), \ldots, \mathbf{E}(o,m))$ are extracted from sensor data for each object o in $\mathbf{S}$. Trajectories $\mathbf{J}(o)$ are made up of object state estimates $\mathbf{E}(o,t)$ for every time step t. $\mathbf{E}(o,t)$ contain absolute poses $\mathbf{T}$ that are acquired using object localizers on stereo images [12], otherwise they are empty. In this setting, we regard recognizing a scene $\mathbf{S}$ at pose $\mathbf{T}_F$ as equal to estimating to which degree an object configuration $\{\mathbf{E}(o,t)\}$, given at time step t, corresponds to a model we learnt for $\mathbf{S}$.

IV. IMPLICIT SHAPE MODEL FOR PASSIVE SCENE RECOGNITION ON SPATIAL RELATIONS

A. Scene model learning

We extend Implicit Shape Models to represent spatial relations between n objects o in a scene $\mathbf{S}$ as introduced in [13]. As an ISM represents a scene $\mathbf{S}$ as a star topology of relations $\{\mathbf{T}_{oF}\}$ towards a common reference, a reference object o_F with pose $\mathbf{T}_F$ is chosen among objects o. Learning such an ISM for scene $\mathbf{S}$ is accomplished by inserting entries into a common table. An entry is added for each estimate $\mathbf{E}(o,t)$ in trajectories $\mathbf{J}(o)$ that belong to objects o of scene $\mathbf{S}$. Besides identifiers for scene and object, each entry contains two relative poses: $\mathbf{T}_{Fo}(t)$ transforms from the coordinate frame of object o into the frame of reference object o_F and $\mathbf{T}_{oF}(t)$ represents a transformation from reference o_F towards object o. Some training data is shown in Fig. 2 on the left. To express how much recognition of scene $\mathbf{S}$ depends on having detected object o, we add beliefs $b_{\mathbf{S}}(o)$ for every o to our scene model. Beliefs $b_{\mathbf{S}}(o)$ are initialized with an equal distribution on $\{o\}$ or using the number of entries for o and $\mathbf{S}$ divided by the number of all entries for $\mathbf{S}$.

B. Passive scene recognition

For a given set of input objects $\{i\}$ that are detected at poses $\{\mathbf{T}\}$, scene recognition with ISMs, being learnt for scenes $\{\mathbf{S}\}$, takes place as a voting process. Each input object i casts votes $\mathbf{v}$ where it expects references of scenes $\mathbf{S}$ based on its pose $\mathbf{T}$. A vote $\mathbf{v}$ contains an estimation from object i for the absolute pose $\mathbf{T}_F \leftarrow \mathbf{T} \cdot \mathbf{T}_{oF}$ of scene reference o_F as well as the pose $\mathbf{T}_{Fo}$ of object i relative to reference o_F. Both $\mathbf{T}_{oF}$ and $\mathbf{T}_{Fo}$ result from an entry in the ISM table that matches scene $\mathbf{S}$ and object i. Despite a vote $\mathbf{v}$ contains poses in six DoF, it is casted on position $\mathbf{p}_F$ of reference o_F, i.e. in position space $\mathbb{R}^3$, to reduce time and memory that voting requires. To be able to cumulate votes, we subdivide $\mathbb{R}^3$ into a voxel grid accumulator $\mathbf{B}_{\mathbf{S}}$ for every scene $\mathbf{S}$.

After voting is finished, an exhaustive search on accumulator $\mathbf{B}_{\mathbf{S}}$ is performed to detect hypotheses about existing instances of scene $\mathbf{S}$. During this process, a greedy search considering six DoF poses is executed on every grid element

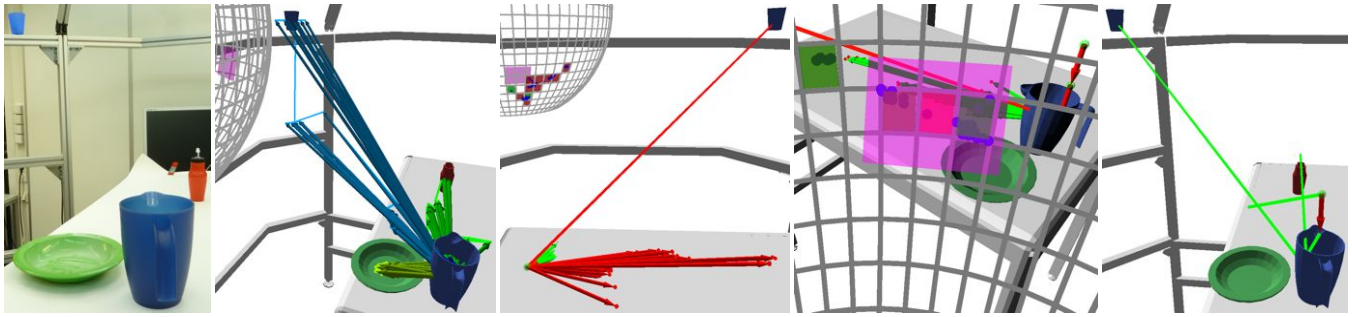

Fig. 2. Left: Scene in which blue cup is moved from table to frame. Center-left: Arrow-shaped relative object poses in ISM training data. Center: First search step - Best scene hypothesis for detected cup points to table and predicts positions of other objects. Center-right: Second search step - After two objects detected, remaining position votes projected on PTU sphere with quads colored according to counters & NBV in purple. Right: Recognition result.

$\mathbf{B_S}(X,Y,Z)$ individually. Among the objects for which votes are present in grid element $\mathbf{B_S}(X,Y,Z)$, it seeks a set $\{i\}_\mathbf{S}$ of maximum size, in which an arbitrary object $i \in \{i\}_\mathbf{S}$ permits to predict absolute poses $\mathbf{T}'_p = \mathbf{T}_F \cdot \mathbf{T}'_{Fo}$ of all other objects $i' \in \{i\}_\mathbf{S}$ sufficiently well. $\mathbf{T}_F$ results from a vote of i in $\mathbf{B_S}(X,Y,Z)$ for the pose of the reference. Beyond acquiring confidence about nothing but the existence of a scene $\mathbf{S}$ through voting, this approach enables us identify which objects cause a specific recognition result. Largest input object sets $\{i\}_\mathbf{S}$ are rated by aggregating beliefs $b_\mathbf{S}(i)$ of all objects $i \in \{i\}_\mathbf{S}$. In case, object set $\{i\}_\mathbf{S}$ is rated as sufficiently good, it is returned as a recognized instance $\mathbf{I_S}$ of scene $\mathbf{S}$ with reference pose $\mathbf{T}_F$ und its rating $j(\mathbf{I_S})$.

V. IMPLICIT SHAPE MODEL HIERARCHY BY AGGLOMERATIVE CLUSTERING

A. Scene model learning

Up to this point, Implicit Shape Models describe scenes in terms of spatial relations of n objects $\{o\}$ towards a common reference o_F. In Fig. 1, a scene is depicted in which non-reference objects $o \in \{o\} \setminus o_F$ are distinctly related to each other. As such relations are not considered by the ISM, their violation is not detected, leading to false positive scene detections. Our solution [13] to this problem is dividing object set $\{o\}$ into subsets based on spatial relations detected in $\{o\}$. ISMs are trained on each subset separately. Hierarchical agglomerative clustering is used to build a binary tree of ISMs $\{m\}$. Leafs $\{o\}_L$ in tree $\{m\}$ stand for real objects for which trajectories $\mathbf{J}(o)$ were recorded, while inner nodes $\{o\} \setminus \{o\}_L$ represent the references $\{o_F\}$ of the ISMs in tree $\{m\}$. ISMs $m \in \{m\}$ pass their results as input to adjacent ISMs $m' \in \{m\}$ on the next lower level in $\{m\}$ except for ISM m_R. Reference o_F of m is handled as a regular object in m'. As the reference of ISM m_R is the root o_R of tree $\{m\}$, it returns the actual scene recognition results.

The linkage criteria used in our clustering approach are heuristics $\{H\}$ that rate relations between pairs of objects $o_j, o_k \in \{o\}$. To cope with the problem shown in Fig. 1 on the left, we developed the heuristic $H_C(o_j, o_k)$. It analyzes temporal continuity in the direction of two types of vectors: $\mathbf{p}_{jk}(t)$ connecting positions of objects o_j and o_k as well as $\mathbf{q}_{jk}(t)$ which is their relative orientation. For a pair of trajectories $\mathbf{J}(o_j)$ and $\mathbf{J}(o_k)$, both of length l, discontinuities u are determined by repeatedly comparing angles $\angle(\mathbf{p}_{jk}(t), \mathbf{p}_{jk}(t+x))$ resp. $\angle(\mathbf{q}_{jk}(t), \mathbf{q}_{jk}(t+x))$ between the same vectors, valid at different time steps, to a given threshold. Beyond representing distributions of values of $\mathbf{p}_{jk}(t)$ and $\mathbf{q}_{jk}(t)$, H_C captures how both change over time by counting discontinuities. For $\mathbf{p}_{jk}(t)$, H_C is defined as:

$n \leftarrow 1$
for $t \leftarrow 1 \ldots l$ **do**
  if $E(o_j,t) \neq \emptyset$ in $\mathbf{J}(o_j) \wedge E(o_k,t) \neq \emptyset$ in $\mathbf{J}(o_k)$ **then**
    $\mathbf{p}_{jk}(n) \leftarrow \mathbf{T}_j(t)^{-1} \cdot \mathbf{p}_k(t)$ and $n \leftarrow n+1$
if $n > \varepsilon \cdot l$ **then**
  $x \leftarrow 1$ and $u \leftarrow 0$
  for $i \leftarrow 1 \ldots n-1$ **do**
    while $i+x \leq n \wedge \angle(\mathbf{p}_{jk}(i), \mathbf{p}_{jk}(i+x)) < d$ **do**
      $x \leftarrow x+1$
    if $i+x \leq n$ **then**
      $u \leftarrow u+1$
    $i \leftarrow i+x$, and $x \leftarrow 1$
  return $1 - u/l$

Agglomerative clustering unifies pairs of trajectories $\mathbf{J}(o_j)$, $\mathbf{J}(o_k)$ to clusters m until all heuristics H return ratings beneath threshold e. The remaining trajectories $\mathbf{J}(o)$ are subsumed to root ISM m_R. This process functions as follows.

$(H_M, o_M, q_M) \leftarrow \underset{(H,o,q)\in(\{H\},\{o\},\{o\})}{\operatorname{argmax}} H(o,q)$
while $H_M(o_M, q_M) > e$ **do**
  Learn ISM m with $\mathbf{J}(o_M)$, $\mathbf{J}(q_M)$. o_F taken among o_M, q_M
  $\{\mathbf{J}(o)\} \leftarrow \{\mathbf{J}(o)\} \setminus (\mathbf{J}(o_M) \cup \mathbf{J}(q_M))$
  $\{\mathbf{J}(o)\} \leftarrow \{\mathbf{J}(o)\} \cup \mathbf{J}(o_F)$
  $(H_M, o_M, q_M) \leftarrow \underset{(H,o,q)\in(\{H\},\{o\},\{o\})}{\operatorname{argmax}} H(o,q)$
  $\{m\} \leftarrow \{m\} \cup m$
Learn root ISM m_R with $\{\mathbf{J}(o)\}$
$\{m\} \leftarrow \{m\} \cup m_R$

B. Passive scene recognition

In the first step of scene recognition with a tree of ISMs $\{m\}$, inputs objects $\{i\}$, given at poses $\{\mathbf{T}\}$, cast votes in every ISM m, where they match a leaf node $\{o\}_L$. In tree $\{m\}$, beliefs $b_\mathbf{S}(o)$ of leaf nodes $\{o\}_L$ are set to 1, while a belief $b_\mathbf{S}(o)$ of 0 is assigned to every inner node $\{o\} \setminus \{o\}_L$.

Fig. 3. Left two images: Objects on a common plate used in Sec. VIII-B and object trajectories resp. relative object poses while rotating plate during training. Else: PTU observing scene used in Sec. VIII-C with objects distributed across laboratory. Learning data for both scenes in setup is on the right.

After a first recognition run in every ISM m as described in Sec. IV-B, returned scene references o_F may only be expected from those ISMs m that contain leafs $\{o\}_L$. Inner nodes $\{o\} \setminus \{o\}_L$ do not participate in voting yet, leading to missing or incomplete recognition results. The ratings of those recognition results are used to update beliefs $b_{\mathbf{S}}(o_F)$ in the tree and references o_F are added to the object set $\{i\} = \{i\} \cup \{o_F\}$. We repeat the process, described until this point, to iteratively generate new reference objects o_F, thereby increasing potential input to ISMs in tree $\{m\}$ and beliefs $b_{\mathbf{S}}(o_F)$ for present scene references. We stop propagating beliefs $b_{\mathbf{S}}(o)$ from the leafs $\{o\}_L$ to the root o_R as soon as $b_{\mathbf{S}}(o)$ converge in tree $\{m\}$. Then we extract recognition results for scenes $\mathbf{S}$ from ISM m_R whose reference is root o_R of $\{m\}$. An example is depicted in Fig. 2 on the right.

VI. QUADGRID AND VIEWS

Suppose a sensor setup able to localize objects is mounted on a motorized Pan-Tilt Unit (PTU). As a simplification we assume that this setup is located at the center of a frame rotating in two degrees of freedom ρ and τ. In this case, we can model all possible views $V(u,v)$ of this setup on the environment as lying on a sphere around it as depicted in Fig. 2. Ignoring its radius, points on this sphere can be described by angles θ and ϕ, which we assume to be identical to the angle positions ρ and τ of the PTU. As the workspace of the PTU is limited, only a subset $[\rho_{\min}, \rho_{\max}] \times [\tau_{\min}, \tau_{\max}]$ of the possible views $V(u,v)$ can be reached. We subdivide this subset into a two-dimensional grid $Q = [-m,m] \times [-n,n]$ similar to [11] in order to accumulate votes $\mathbf{v}$ on positions $\mathbf{p} \in \mathbb{R}^3$ in it and call it quadgrid. We assign to each quad at position $(u,v) \in Q$ a counter $c(u,v)$ for votes $\mathbf{v}$ that fall into it after having been projected on the sphere. For each quad at (u,v), we define a flag $e(u,v,o)$ telling whether we searched for object $o \in \{o\}$ in it. We define views $V(u,v) = [-a+u, u+a] \times [-b+v, v+b]$ at positions $(u,v) \in Q$ ($V = [-a,a] \times [-b,b]$) as being aligned to the quadgrid.

VII. RELATION-BASED OBJECT SEARCH FOR ACTIVE SCENE RECOGNITION

A. Object Search

Until this point, we assumed that a fixed input set of objects $\{i\}$ is given for recognizing a set of scenes $\{\mathbf{S}\}$ that are made up of objects $\{o\}$, with ISM tree $\{m\}$. An alternative to acquire localization results actively is spiral search [14], an uninformed search strategy for points in a plane. Starting at a given point (u_0, v_0), it traverses the plane of quadgrid Q on a spiral track until the point, being looked for, is found. With no objects given in advance, we use this strategy on quadgrid Q to find at least one object i that is mandatory for our active scene recognition approach to start searching based on spatial relations. We assume that the ISM tree $\{m\}$ is learnt as described in previous sections but with the sensor setup being guided by hand to capture scenes across different viewpoints during learning. Based on the set $\{o\}_A \subset \{o\}$ of already detected objects, we use the passive scene recognition approach presented in this paper to predict possible positions of objects $\{o\}_P \subset \{o\}$, not detected yet. This system runs in a loop, in which views V are successively being switched until all objects $o \in \{o\}$ are found or no promising hypotheses for potential object locations remain. Given the view V_C, to which the sensor setup currently points, the body of this loop is defined as follows.

Perform object localization for $o \in \{o\}_P$ on view V_C.
if we found an object $o \in \{o\}_P$ **then**
  $\{o\}_A \leftarrow \{o\}_A \cup o$, $\{o\}_P \leftarrow \{o\}_P \setminus o$ and $\{\mathbf{I_S}\} \leftarrow \emptyset$
  $\{\mathbf{I_S}\} \leftarrow$ results of hierarchical ISM recognition on $\{\mathbf{S}\}$ & $\{o\}_A$
  if $\{o\}_P = \emptyset$ **then**
    for all $\mathbf{S} \in \{\mathbf{S}\}$ **do**
      Add to results: $\underset{\mathbf{I_S} \in \{\mathbf{I_S}\}}{\operatorname{argmax}} j(\mathbf{I_S})$ using rating $j()$
    Quit loop.
  for all $(u,v) \in Q$ **do**
    $c(u,v) \leftarrow 0$ and $e(u,v,o) \leftarrow$ **true**
  $\mathbf{I}_{\max} \leftarrow \underset{\{\mathbf{I_S} | \mathbf{I_S} \in \{\mathbf{I_S}\} \wedge \mathbf{S} \in \{\mathbf{S}\}\}}{\operatorname{argmax}} j(\mathbf{I_S})$
  calcPositionHypotheses($\mathbf{I}_{\max}, \mathbf{T}_{\max}, 1$)
for all $(u,v) \in V_C$ **do**
  $c(u,v) \leftarrow 0$ and $\forall o \in \{o\}_P : e(u,v,o) \leftarrow$ **true**
Estimate next-best-view V_N with rating $r(V_N)$ based on V_C
while $r(V_N) < d$ **do**
  if $|\{\mathbf{I_S}\}| > 1$ **then**
    $\{\mathbf{I_S}\} \leftarrow \{\mathbf{I_S}\} \setminus \mathbf{I}_{\max}$
    calcPositionHypotheses() on $\mathbf{I}_{\max}$, recalculated on $\{\mathbf{I_S}\}$
    Estimate V_N with rating $r(V_N)$ based on V_C
  else
    for all $\mathbf{S} \in \{\mathbf{S}\}$ **do**
      Add to results: $\underset{\mathbf{I_S} \in \{\mathbf{I_S}\}}{\operatorname{argmax}} j(\mathbf{I_S})$
    Quit loop.
Move PTU to V_N.

As soon as a previously not found object is detected and scene recognition with hierarchical ISMs is run to

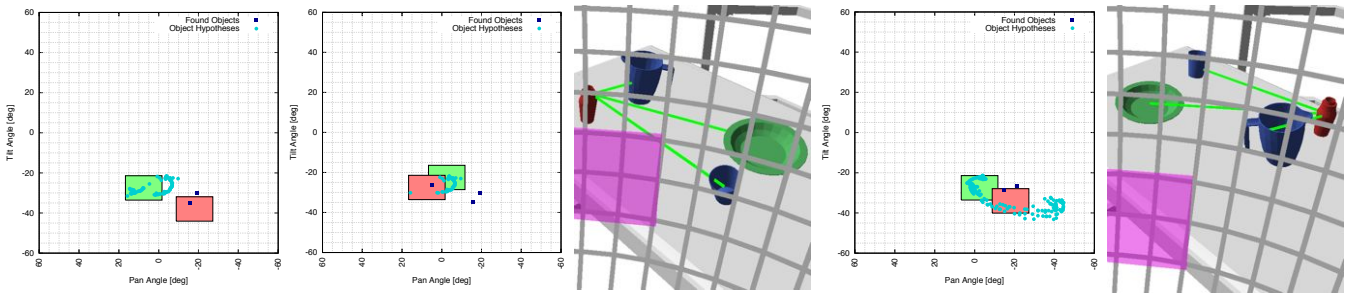

Fig. 4. Search steps on quadgrid and recognition results for rotating-scene scenario. Quadgrid shown as two-dimensional grid with a resolution of five degrees. Current view of sensors shown as a red box, while next-best-view drawn in green. Hypotheses on positions used for NBV shown in turquoise.

update our hypotheses about present scenes, we greedily choose the scene hypothesis $\mathbf{I}_{\max}$, currently rated best and of any considered scene $\mathbf{S} \in \{\mathbf{S}\}$. Assuming that $\mathbf{I}_{\max}$ is valid, we use a voting scheme to calculate the view V_N, the sensor setup shall be pointed to, next. The first step of this process generates votes where to expect objects, in the method calcPositionHypotheses(), after which V_N is chosen by interpreting casted votes in a next-best-view estimation step. This process is depicted in Fig. 2. To prevent $(u,v) \in Q$ being searched multiple times for the same objects o, we mark it as explored using $e(u,v,o)$. While objects are searched using hypothesis $\mathbf{I}_{\max}$, we invalidate counters of object position hypotheses $c(u,v)$ as we traverse views V. When hypothesis $\mathbf{I}_{\max}$ delivers no more promising views V_N to move to, we switch to the next-best hypotheses, possibly for another scene $\mathbf{S}'$ that is being looked for as well.

B. Voting on Object Position Hypotheses

for all $o \in \{o\}$ of m **do**
  if $o \in \{o\}_P$ **then**
    Extract all ISM table entries matching m and o
    for all matching table entries **do**
      Extract $\mathbf{T}_{Fo}$ from entry
      $\mathbf{T} \leftarrow \mathbf{T}_F \cdot \mathbf{T}_{Fo}$ with $\mathbf{T}_F$ given for m
      if $\mathbf{T} \approx \mathbf{T}_F$ **then**
        $w \leftarrow w_F \cdot$ number of matching table entries
        if $o \in \{o\}_L$ **then**
          Get $(u,v) \in Q$ of $\mathbf{p}$ from $\mathbf{T}$ (Projection on sphere)
          if $e(u,v,o) =$ **false then**
            $c(u,v) \leftarrow c(u,v) + w$
        else
          calcPositionHypotheses$(o, \mathbf{T}, w)$
        Quit loop.
      else
        if $o \in \{o\}_L$ **then**
          Get $(u,v) \in Q$ of $\mathbf{p}$ from $\mathbf{T}$ (Projection on sphere)
          if $e(u,v,o) =$ **false then**
            $c(u,v) \leftarrow c(u,v) + w_F$
          else
            calcPositionHypotheses$(o, \mathbf{T}, w_F)$

Given reference object o_F of scene m with its pose $\mathbf{T}_F$ and a weight w_F, hypotheses about absolute positions $\mathbf{p}$ of objects $o \in \{o\}_P$ are calculated in calcPositionHypotheses() by using entries in the ISM table learnt for m. By using poses $\mathbf{T}_{Fo}$ of object o relative to reference o_F, we reverse the principle used in ISMs for voting on poses of scene references. Once $p \in \mathbb{R}^3$ is estimated, we project in onto the sphere and determine into which quad at position $(u,v) \in Q$, it falls. Counter $c(u,v)$ is incremented accordingly. Both is shown in Fig. 2 in the center. As we use hierarchical ISMs, such statements only apply to leaf nodes $\{o\}_L$. If object o is an internal node, we pass every hypothetical pose $\mathbf{T}$ having been estimated for o to the ISM in the next-higher level in the tree $\{m\}$ that uses o as reference. Positions $\mathbf{p}$ are calculated for every combination of pose votes on the path from root node o_R to leaf $\{o\}_L$. To reduce computations, we introduce weights w that permit to vote once with a weight instead of calculating redundant votes. This is important since half of the nodes in $\{m\}$ vote on references being identical to them.

C. Next-best-view estimation

Given the current view V_C at position $(u,v)_C \in Q$ and Q filled with votes for objects $o \in \{o\}_P$, we estimate the next view V_N to move to in a greedy manner using a sliding window approach. This is done by maximizing a reward $r(u,v)$ that consists of a cost term $r_c(u,v)$ and $r_d(u,v)$ which expresses the confidence to find any objects $o \in \{o\}_P$ in a view located in (u,v). Costs $r_c(u,v)$ are defined as normalized absolute distances on each joint of a kinematic chain similar to [2], but defined for the PTU. $r_d(u,v)$ is acquired by summing all object positions counts in the window on quadgrid Q that a view located at (u,v) defines.

$r_Q \leftarrow \max\left(1, \sum_{(u,v)\in Q} c(u,v)\right)$
for $u \leftarrow -m+a \ldots m-a$ **do**
  for $v \leftarrow -n+b \ldots n-b$ **do**
    $r_c(u,v) \leftarrow \sum_{(i,j)\in V} c(u+i, v+j)$ and $r_c(u,v) \leftarrow {}^{r_c(u,v)}/_{r_Q}$
    $r_d(u,v) \leftarrow 0.5 \cdot \frac{|u_C - u|}{2(m-a)+1} + 0.5 \cdot \frac{|v_C - v|}{2(n-b)+1}$
    $r(u,v) \leftarrow a_1 \cdot r_c(u,v) + a_2 \cdot (1 - r_d(u,v))$
return $(u,v)_N \leftarrow \underset{(u,v)\in[-m+a,m-a]\times[-n+b,n-b]}{\operatorname{argmax}} r(u,v)$

VIII. EXPERIMENTS AND RESULTS

A. Experimental setups

Exemplary runs of the presented active scene recognition are analyzed for three hierarchical scene models. Data from demonstrations, out of which these models are learnt, is visible in Fig. 3. The size of view V during object search

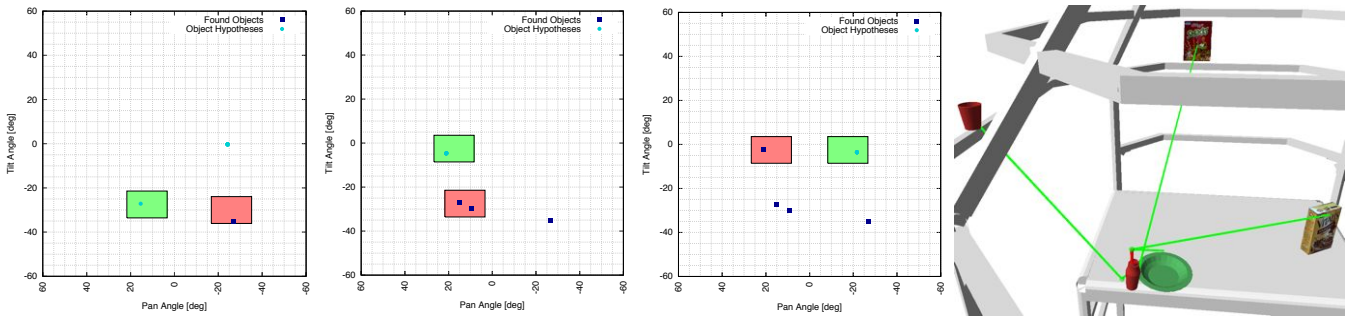

Fig. 5. Search steps on quadgrid and recognition results for scenario with two scenes. On the right: Relations in ISM recognition results shown as lines, that turn from green to red as ratings of recognition results decrease. They meet at a reference shown as green sphere. Lines from reference object o_F to ISM reference are invisible as both are equal. Red arrows stand for ISM references being passed as objects to other ISMs that reside in lower tree levels.

is set to half of the viewing angle of the sensor setup since object localization quality decreases considerably at the borders of the processed images. On the two left images, a configuration of four objects is shown which all stand on a plate on a table. On the right, two different configurations reside in the same setting, having one object, a ketchup bottle, in common. Experiments are run on a PC with a "Core i5 750" and 4 GB RAM. Objects are localized using [12].

B. Recognizing scenes indepedent of their emplacement

The experiments, conducted in this section, illustrate how active scene recognition based on spatial relations is independent of absolute poses of detected objects. Over the object search process, it is shown how passive scene recognition with hierarchical ISMs deals with incomplete input data. During learning, all objects on the plate are first moved in proximity of their final locations on the plate, before the plate is rotated about 180° on the table. All objects are subsumed to a common cluster, since heuristic H_C detects no specific relations. The trained ISM table contains 472 entries.

We performed two object search runs with the scene being rotated about 180° in-between. In both cases the sensor setup initially points at the pair of objects being on the left. Both scenarios are shown in Fig. 4. The first run, to which the three pictures on the left of Fig. 4 belong, reaches two additional views, before the scene is detected completely by our hierarchical ISM as shown in the middle of Fig. 4. In the initial view, shown in red on the left, we find both cup and plate. The confidence of the scene hypothesis $\mathbf{I_S}$ rated best, lies at $j(\mathbf{I_S}) = 0.5$. The NBV V_N calculated based on this hypothesis, is shown in green. Compared to other views, containing votes for potential object positions, its distance to the current view V_C is high, but its confidence reward $r_d(u,v) = 0.69$ compensates. NBV V_N contains hypotheses for both ketchup bottle and measuring both being looked for. As visible next to the right, unexpectedly only ketchup is localized in this view. Based on an updated best scene recognition result $\mathbf{I_S}$ with a confidence of $j(\mathbf{I_S}) = 0.75$, the measuring cup is searched in a NBV V_N with confidence reward $r_d(u,v) = 0.69$, solely processing votes for this object. It is found there and the scene is entirely detected. During this run, NBVs are calculated two times with an average runtime of 39.08 ms. Estimating hypotheses of potential object positions is done twice and takes in average 4.22 ms. Passive scene recognition takes 0.87 ms.

The second search run takes only one view in addition to the initial. Votes as depicted in the second image from the right in Fig. 4 differ from the first run, since other objects, i.e. measuring cup and ketchup bottle are found in the initial view. As position hypotheses are more spread in that run, not the confidence reward $r_d(u,v) = 0.24$, but low travel costs $r_c(u,v) = 0.05$ are decisive for choosing NBV V_N. In NBV V_N, we find both missing objects and the scene is entirely detected as shown on the right in Fig. 4. The runtime for NBV estimation is 38.30 ms, position hypothesis estimation takes 2.92 ms and ISMs 0.24 ms. Despite objects being switched in front of the camera, active scene recognition moves in similar directions when choosing a NBV. This shows processing spatial relations allows to adapt knowledge about object poses in contrast to using absolute object poses.

C. Recognizing scenes independent of their number

In this experiment with two scenes present at once, objects are not moved during training. As continuity in H_C only depends on detection noise, objects being closer to each other are subsumed to clusters. For example in one scene, which we call B, the ketchup bottle and a plate are subsumed. The reference of the cluster is subsumed with a cup. We denote the other scene as A. There are 68 votes in the ISM table.

Starting with a view on a vitalis cornflakes box shown on the bottom right in the right picture in Fig. 5, three additional views are necessary to detect both scenes completely. Using a detection result for vitalis, our best scene recognition result $\mathbf{I_A}$ has a confidence of $j(\mathbf{I_A}) = 0.33$ and belongs to scene A. Since all other objects of A did not move during training the scene model, views on them are rated equally by rewards $r_d(u,v)$. This situation is shown on the left in Fig. 5. Therefore NBV V_N is chosen based on travel costs $r_c(u,v)$, moving the sensor setup to the ketchup bottle. Due to the working area of the pan angle being larger, travel costs to it are lower than to the Smacks box. Unexpected for the active scene recognition, an additional plate is found in view V_N. As both ketchup bottle and plate belong to scene B, our best hypotheses $\mathbf{I_B}$ receives a confidence of $j(\mathbf{I_B}) = 0.66$.

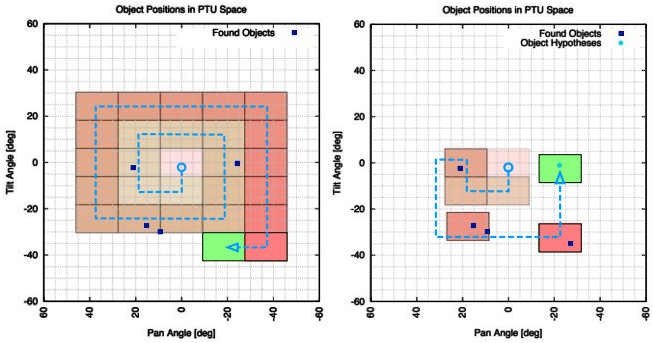

Fig. 6. All views visited by spiral search on the left and by relation-based object search on the right. Their order is illustrated by blue arrows.

For Scene *A*, a recognition result $\mathbf{I_A}$ with $j(\mathbf{I_A}) = 0.66$ is provided. The hypothesis for scene *B* is selected at random, since both ratings are equal. Now searching for scene *B* instead of scene *A*, the NBV V_N points towards votes for the cup as shown in the second plot from the left in Fig. 5. As this object is found and a recognition result $\mathbf{I_B}$ for scene *B* with rating $j(\mathbf{I_B}) = 1$ is available, all hypotheses for scene *B* are excluded from further processing and scene *A* is considered again. In the second figure from the right, a NBV is depicted, illustrating that the view changes from the cup to the smacks box. Afterwards both scenes are fully detected. In this run, NBV estimation is executed four times with an average runtime of 46.39 ms. Position hypotheses calculation takes 1.08 ms and hierarchical ISM detection 0.24 ms.

D. Efficiency of Scene Recognition

We ran the presented active scene recognition approach against a conjunction of passive ISM recognition and spiral search, starting both at view $V(0,0)$ in the scenario described in the previous section. We intend to compare object search based on spatial relations, after an initialization with spiral search, to uninformed search by itself. As shown in Fig. 6, relation-based search took 6 additional views to find both scenes, while spiral search took 26. Movement costs of relation-based search were 59° in pan direction and 86° in tilt, while spiral search took 202° in pan and 182° in tilt. These were the significant runtime factors during search.

We analyzed runtimes for passive scene recognition with hierarchical and non-hierarchical ISMs. As shown in Fig. 7, we measured runtimes for different sizes of input object sets. Both systems run at most 10 ms with six input objects. Increasing learning data set size does almost not affect non-hierarchical ISMs, while hierarchical ISMs run with linear time on the lengths of recorded object trajectories. A similar observation holds for the number of input objects.

IX. CONCLUSIONS AND FUTURE WORKS

An approach has been presented that searches for objects based on spatial relations in order to improve its estimation of scenes, present in its environment. Scenes are modeled as configurations of objects in terms of six DoF interrelationships using Implicit Shape Models. Models are acquired from

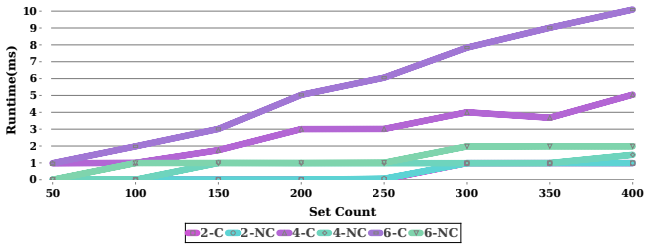

Fig. 7. Passive scene recognition runtimes for 2 to 6 input objects resp. hier. (C) or non-hier. (NC) ISMs shown for different trajectory lengths.

demonstrations by humans. To overcome limitations of ISMs in modeling relations in scenes, we generate hierarchies of ISMs using agglomerative clustering. Object search is done by successive estimation of next-best-views based on votes for potential positions of not-found objects in a quadgrid. Voting is realized as a reversion of scene detection with ISMs. Starting with the objects already detected, object positions are calculated using object relations, starting from the location of the scene hypothesis rated best at this point. Experiments show that scenes can be found independent of their emplacement or their number using a real-time capable passive scene recognition. Object search using spatial relations outperforms uninformed search like spiral search. Future work includes processing votes on hypothetical object positions separately for each object in order to model the influence of object identities on next-best-view estimation.

REFERENCES

[1] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition*, 2009.
[2] R. Eidenberger, T. Grundmann, M. Schneider, W. Feiten, M. Fiegert, G. v. Wichert, and G. Lawitzky, "Scene analysis for service robots," in *Towards Service Robots for Everyday Environments*. Springer, 2012.
[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. Journal of CV*, 2001.
[4] A. Borji and L. Itti, "Scene classification with a sparse set of salient regions," in *Int. Conf. on Robotics and Automation*, 2011.
[5] T. Southey and J. Little, "3d spatial relationships for improving object detection," in *Int. Conf. on Robotics and Automation*, 2013.
[6] A. Ranganathan and F. Dellaert, "Semantic modeling of places using objects," in *Robotics: Science and Systems Conference*, 2007.
[7] D. Joho, G. Tipaldi, N. Engelhard, C. Stachniss, and W. Burgard, "Nonparametric bayesian models for unsupervised scene analysis and reconstruction," in *Robotics: Science and Systems Conference*, 2012.
[8] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. Journal of CV*, 2008.
[9] S. Kriegel, C. Rink, T. Bodenmuller, A. Narr, M. Suppa, and G. Hirzinger, "Next-best-scan planning for autonomous 3d modeling," in *Int. Conf. on Intelligent Robots and Systems*, 2012.
[10] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," in *Int. Conf. on Intelligent Robots and Systems*, 2011.
[11] Y. Ye and J. K. Tsotsos, "Sensor planning for 3d object search," *Computer Vision and Image Understanding*, 1999.
[12] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6d object localization for grasping with humanoid robot systems," in *Int. Conf. on Intelligent Robots and Systems*, 2007.
[13] P. Meißner, R. Reckling, R. Jäkel, S. R. Schmidt-Rohr, and R. Dillmann, "Recognizing scenes with hierarchical implicit shape models based on spatial object relations for programming by demonstration," in *Int. Conf. on Advanced Robotics*, 2013.
[14] E. Langetepe, "On the optimality of spiral search," in *ACM-SIAM Symposium on Discrete Algorithms*, 2010.