Temporal Integration of Feature Correspondences For Enhanced Recognition in Cluttered And Dynamic Environments

Thomas Fäulhammer, Aitor Aldoma, Michael Zillich and Markus Vincze

***Abstract*— We propose a method for recognizing rigid object instances in RGB-D point clouds by accumulating low-level information from keypoint correspondences over multiple observations. Compared to existing multi-view approaches, we make fewer assumptions on the recognition problem, dealing with cluttered and partially dynamic environments as well as covering a wide range of objects. Evaluation on the publicly available TUW and Willow datasets showed that our method achieves state-of-the-art recognition performance for challenging sequences of static environments and a significant improvement for environments partially changing during the observation.**

I. INTRODUCTION

The detection of objects and the estimation of their position and orientation in challenging environments is a key capability for robotic agents with manifold applications in industrial and service robotics [1], [2]. Despite recent progress (e.g. [3]–[6]), algorithms deployed for object recognition are still unable to perform robustly in scenarios where objects undergo strong occlusions from the current vantage point or are viewed from ambiguous, uninformative perspectives. For example, recognition methods relying on features that exploit texture information might be unable to detect partially textured objects whenever the informative parts are not visible from the current position of the robot.

However, in the robotics context, where agents are able to actively explore the environment, it is possible to reposition the sensing device mounted on the robot in order to seek for advantageous vantage points (with respect to the recognition capabilities of the underlying recognition method) as well as to mitigate problems caused by occlusions. While these novel viewpoints offer additional information that potentially enhance object recognition, it is beneficial to consider the gathered information of the environment up to the current observation and detect objects in this *multi-view* setting instead of considering each vantage point as an independent piece of information. The benefits of the multi-view setting are depicted in a toy example in Fig. 1. While single observations often do not present enough correspondences to construct a hypothesis for an object, merging individual correspondences from multiple observations can overcome this lack of information and construct hypotheses even in highly cluttered and occluded environments.

Despite the great potential of the multi-view setting discussed above, it assumes that the environment being explored is static. While this assumption might hold during short periods of time or for parts of the scene, it is unrealistic to expect that environments remain static over larger periods of time, specially in environments such as homes or offices populated by people. Indeed, some recent works (e.g. [7], [8]) address the dynamic aspects of certain environments in order to infer spatial relationships of entities that might change over time (e.g. is the wallet still on the office table in front of the monitor? Is the patient's medicine still between the bedside lamp and his spectacles case?). To tackle these problems without human supervision on a robot, it is required to have an accurate and efficient visual recognition method adaptable to changes of the environment. This can be achieved by a trade-off between a high recognition rate obtained by static multi-view recognition methods (e.g. [9], [10]), which accumulate visual information from multiple observations assuming an environment that does not change over time, and single-view methods (e.g. [11]–[13]) only sensing the environment at a given time, which are sensible to dynamic changes but usually have a lower recall of present objects.

In Section III, we propose a method that accumulates low-level information (i.e. features, keypoints and correspon-

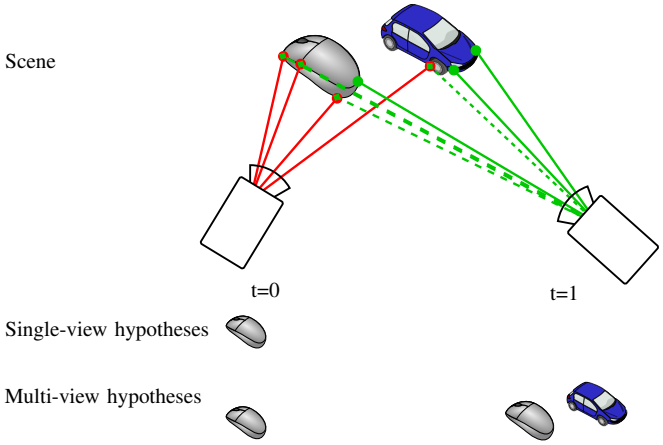


Fig. 1: Keypoints s_0 (red dots) extracted from the scene at time step $t = 0$, matching keypoints from the model database, are transferred into the more recent point cloud at $t = 1$. Rays tracing keypoints to their projection center are depicted as solid lines if they are extracted by a single-view recognizer and as dashed lines if they are extended from previous views. A hypothesis for an object can be created if there are at least three different correspondences between keypoints from the scene and the respective model.

Thomas Fäulhammer, Aitor Aldoma, Michael Zillich and Markus Vincze are with the Vision4Robotics group (ACIN - Vienna University of Technology), Austria {faeulhammer, aldoma, zillich, vincze}@acin.tuwien.ac.at

dences) for multiple observations of a region of interest, which is merged online (i.e. at each sampling time) to enhance performance for a wide range of recognition problems such as in dynamic and cluttered environments. By making fewer assumptions on the scene, we show in Section IV that our method achieves state-of-the-art recognition performance for static environments and a significant better precision/recall rate for slightly changing scene configurations compared to other methods.

II. RELATED WORK

Collet and Srinivasa [14] proposed an efficient multi-view approach for object recognition and pose estimation by a multi-step optimization. Consistent hypotheses, generated over multiple views individually processed by a single-view algorithm and transferred to a common coordinate system, are globally refined using a reduced generalized image with points consistent across all images. While their method avoids the expensive correspondence grouping stage over a possible big set of correspondences merged from multiple views, it can only detect objects that are correctly recognized by the single-view recognizer in at least one of the images. In fact, their multi-view algorithm requires that a pose is seen by at least two-views, and that at least 50% of the points from the different hypotheses are consistent with the final pose to refine a hypothesis. Instead of using a static camera rig, our approach also differs in the observation method of the scene. Our robot is not only able to cover a wider field of view than a rigid camera setup, but also senses the environment at different times. This requires once to deal with changing single-view correspondences, but also enables us to detect dynamic properties of the scene.

Other multi-view systems (e.g. [15], [16]) make similar restrictive assumptions about the environment (i.e. static scene or observed at the same time with multiple cameras). Vikstén *et al* [16] accumulated multiple pose estimates over different temporal and algorithmic cues by a weighted averaging approach to improve accuracy. Although this is a computationally fast method as it scales linearly with the number of views, it requires the detection of object hypotheses in at least one single-view pose estimate.

The proposed approach is a multi-view object instance recognizer based on the work of Aldoma *et al* [9] (batch version) and Fäulhammer *et al* [10] (online version). Both methods use the outputs of a single-view object recognizer over multiple views and transfer them into a common coordinate system containing a 3D reconstruction of the observed scene. Using a 3D extension of the hypotheses framework proposed by Aldoma *et al* [17], the batch method achieves a recall rate of 93.2% and 99.1% on the TUW and Willow Challenge datasets, respectively. Due to this good recognition results, this method has been used for automatically annotating static multi-view RGB-D object instance recognition datasets. In order to correctly annotate an object present in the scene, these two methods assume that each object has to be correctly detected by the single-view recognizer in at least one frame. In contrast, the method proposed in this paper does not necessarily require this assumption as the information gain from multiple views comes from low-level information (i.e. keypoint correspondences). Merging that information into the current view enables to construct a hypothesis even in cases where there is too little information in the respective single-views to construct one (see Fig. 1). To support object hypotheses in the final 3D verification stage, methods [9] and [10] further require a good registration for all views. Contrary, the method presented in this paper uses only the current view for verification and at least provides the approximate quality of a single-view approach in case of a wrong pose estimate of the camera. Obtaining a good reconstruction of the scene furthermore requires an environment remaining completely static over the whole sequence so that the final batch process verifies the merged hypotheses correctly. We propose a novel method that uses information obtained from multiple views online and verifies hypotheses only against the currently observed point cloud.

III. APPROACH

Given a model database of rigid 3D objects and a set of RGB-D views of a sequence ψ over a time period T_ψ

$$\mathcal{S}^\psi = \left\{ \boldsymbol{S}_t^\psi \dots \boldsymbol{S}_{t-T_\psi}^\psi \right\}, \tag{1}$$

the goal of the proposed method is to detect at any timestamp t all present objects known to the system together with their 6DoF pose with respect to the global coordinate system of the robot.

The following section describes the workflow of the recognition system depicted in Fig. 2.

A. Object Model Database

During an offline stage, objects of interest are learned by the system in a controlled setup (e.g. a table-top). Each object is sensed from different vantage points that are brought into alignment in order to generate a 3D point cloud representing the object of interest. For each view, an array of local features, ξ_ω (i.e., SIFT [18] and SHOT [19]), is extracted at corresponding keypoint locations denoted here by $\boldsymbol{m}_\omega$. The object model database is then represented by

$$\mathcal{M} = \left\{ \boldsymbol{m}_\omega, \xi_\omega \,\middle|\, \omega \in \Omega \right\}, \tag{2}$$

where Ω represents a list of unique object identifiers.

B. Filtering Information

In scenarios, where mobile robots or autonomous agents operate in spacious environments, a system accumulating and processing low-level information over an extended period of time will slow down significantly without filtering data. To reduce the amount of irrelevant information for our recognition task for a particular location and at any given time t, we filter scene observations by creating a semantic map of the environment (see Fig. 3). In our case, these are (off-line) annotated spatial regions of interest for recognizing objects and their position over time. Each semantic sequence ψ independently stores information of feature

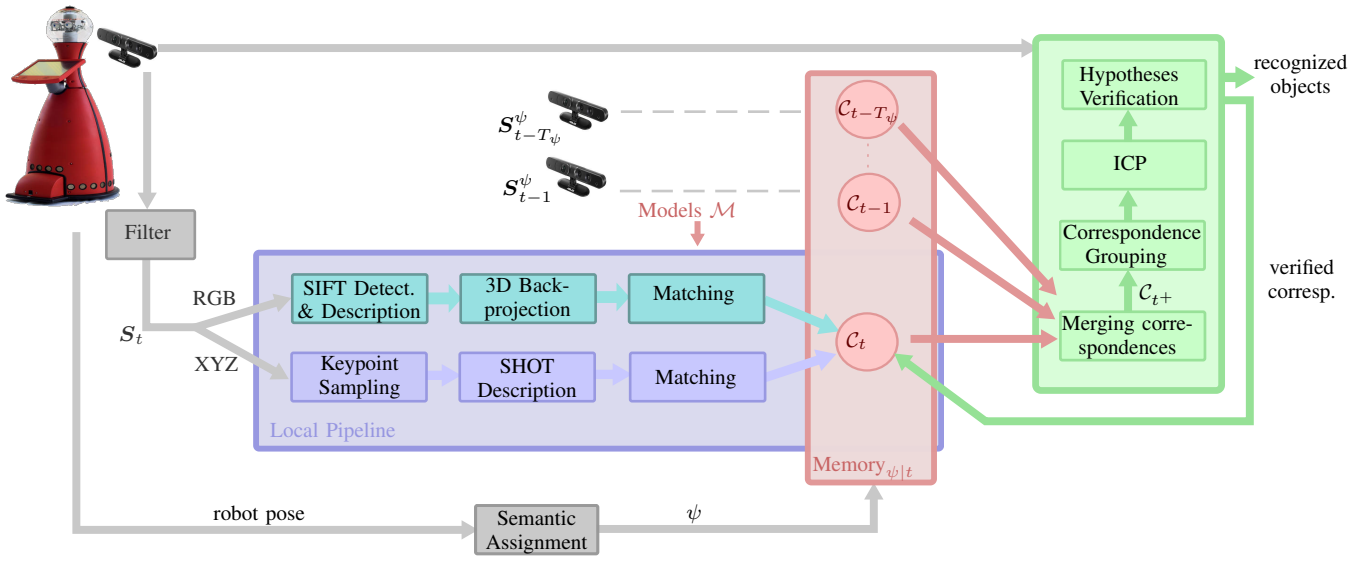


Fig. 2: Workflow of the proposed multi-view method using correspondences $\mathcal{C}_{t^+}$ integrated over a time period T_ψ and multiple observations of a sequence ψ to generate hypotheses of objects and their 3D pose at time t.

correspondences between object models and previous scene observations within sequence ψ. For each timestep t, the system selects the closest sequence ψ (and the corresponding set of observations) based on the current robot pose.

To reduce the number of parameters, we will neglect ψ in the following and only consider a single sequence with a set of point clouds $\mathcal{S} = \{\boldsymbol{S}_t \dots \boldsymbol{S}_{t-T}\}$.

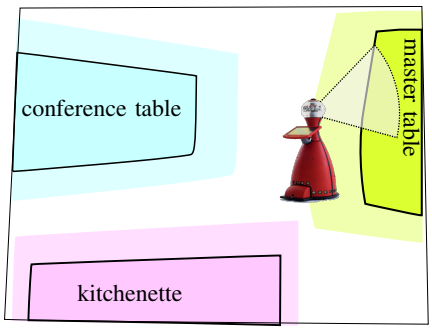


Fig. 3: Representation of an environment with $\psi \in \{$conference table, kitchenette, master table$\}$. Each semantic location ψ stores correspondences information independently. By determining the current location of the robot, the system can disregard information of the *kitchenette* and the *conference table* regions

.

C. Local Pipeline for Single-View Correspondences Extraction

To remove points with high noise level (see [20]), the point cloud of the scene sampled at the current timestamp t is pre-processed by a distance filter with a threshold of 2.5m along the optical axis of the camera. The output of the filter is represented by the point cloud $\boldsymbol{S}_t$. Using L1-norm nearest neighbor (NN) search in the feature domain, the array of keypoints $\boldsymbol{s}_t$ extracted from $\boldsymbol{S}_t$ is matched against model keypoints $\boldsymbol{m}_\omega \, \forall \omega \in \Omega$ via fast approximate indexing (i.e., randomized kd-trees [21]).

Denoting a match for a scene keypoint $\boldsymbol{s}_t^i$ to its first nearest neighbor $\boldsymbol{m}_\omega^j$ by a correspondence $c_{\omega,t}^{i,j} = \left\{\boldsymbol{s}_t^i, \boldsymbol{m}_\omega^j\right\}$, the set of correspondences $\mathcal{C}_{\omega,t}$ at time t and for an object instance ω is represented by

$$\mathcal{C}_{\omega,t} = \left\{ c_{\omega,t}^{i,j} \middle| \boldsymbol{s}_t^i \in \boldsymbol{s}_t \wedge \boldsymbol{m}_\omega^j = \underset{\boldsymbol{m}}{\mathrm{NN}}(\boldsymbol{s}_t^i) \right\}. \tag{3}$$

The system stores the total set of correspondences

$$\mathcal{C}_t = \left\{ \mathcal{C}_{\omega,t} \middle| \omega \in \Omega \right\}, \tag{4}$$

which represents the low-level information about the environment ψ at time t with respect to all models Ω in the database.

D. Merging Correspondences From Multiple Vantage Points

Exploiting information from previously sensed vantage points of the environment, the multi-view recognition approach recursively extends the set of correspondences for its current view C_t by previously stored correspondences C_τ with $\tau < t$. These correspondences between scene and model keypoints are transferred into the current camera coordinate system by a given transform.

To estimate the transform between views, the relative camera pose can be obtained either from the robot pose or calculated from scene keypoints, e.g. using accurate camera tracking algorithms. In environments with few visual features, the transform can also be estimated based on existing object hypotheses $\mathcal{H}$, which, however, requires additional computation time of $\mathcal{O}\left(|\mathcal{H}|^2\right)$ for single-view hypotheses generation and mutual matching [9].

To refine the initial registration of the two point clouds given by any of the estimated transforms mentioned above, the relative camera pose estimate is refined by ICP. For

multiple possible transforms between views, a transform is chosen that minimizes the edge weight function in [9].

As the computational complexity of the following correspondence grouping stage scales by $\mathcal{O}\left(|\mathcal{C}|^2\right)$, it is important to check for redundancy when transferring correspondences into the current camera view. Estimating the pose of an object by a set of corresponding points, redundant information could also state an under-determined problem. Therefore, we iterate through all correspondences beginning with the most current ones and check for redundancy before merging them into a set of correspondences $\mathcal{C}_{\omega,t^+}$ accumulated over a time period T.

Denoting the normal vector of a scene keypoint $\boldsymbol{s}_t^i$ as $\boldsymbol{n}_{s,t}^i$, a correspondence $c_{\omega,\tau}^{i,j} = \left\{\boldsymbol{s}_\tau^i, \boldsymbol{m}_\omega^j\right\}$ is redundant and not extended if the set of accumulated correspondences contains an element $c_{\omega,t^+}^{k,l} = \left\{\boldsymbol{s}_{t^+}^k, \boldsymbol{m}_\omega^l\right\}$ such that

- its scene keypoint transferred into the current view is close to a scene keypoint already in the set,
$$\left\|\boldsymbol{s}_\tau^i - \boldsymbol{s}_{t^+}^k\right\|_2 \leq 0.005 \text{ m}, \tag{5}$$
- the surface normals at the keypoints are approximately aligned,
$$\left(\boldsymbol{n}_{s,\tau}^i\right)^\intercal \boldsymbol{n}_{s,t^+}^k \leq 0.2, \tag{6}$$
- and corresponding model keypoints are in close proximity,
$$\left\|\boldsymbol{m}_\omega^j - \boldsymbol{m}_\omega^l\right\|_2 \leq 0.005 \text{ m}. \tag{7}$$

The accumulated set of correspondences is then obtained by Algorithm 1.

Algorithm 1 Merging correspondences from multiple views

```
init C_{ω|t+} := {}
for τ = 0 → T do
    get redundant correspondences C̃_{ω,τ}
    C_{ω,t+} := { C_{ω,t+}, ( C_{ω,τ} \ C̃_{ω,τ} ) }        ▷ merge
```

E. Generation of Object Hypotheses

As a result of the previous stages, a set of accumulated point-to-point correspondences $\mathcal{C}_{\omega,t^+}$ has been determined. This set of correspondences typically contains outliers that ought to be discarded. As a set of correspondences may comprise several consensus sets related to different instances of a given model in the scene, popular methods for outlier rejection such as RANSAC are not suited to the multi-instance object recognition problem. Hence, specific *correspondence grouping* methods have been devised [22], [23].

We use an extended version of the method proposed in [22], which initializes a seed correspondence and iteratively builds up clusters of correspondences by enforcing geometric consistency between pairs of correspondences $c_{\omega,t}^{i,j}$ and $c_{\omega,\tau}^{k,l}$. Geometric consistency exploits the fact that under rigid body transformations, distances between points are preserved, such that

$$\left|\left\|\boldsymbol{m}_\omega^j - \boldsymbol{m}_\omega^l\right\|_2 - \left\|\boldsymbol{s}_t^i - \boldsymbol{s}_t^k\right\|_2\right| < \varepsilon, \tag{8}$$

where ε represents the maximum allowed difference between the feature distances measured on the keypoints of model and the scene.

In addition, it is possible to enforce an additional constraint based on angle consistency. Let $\boldsymbol{n}_{m,\omega}^j$ be the surface normal at point $\boldsymbol{m}_\omega^j$, we introduce an additional consistency check,

$$\left|\left(\boldsymbol{n}_{m,\omega}^j\right)^\intercal \boldsymbol{n}_{m,\omega}^l - \left(\boldsymbol{n}_{s,t}^i\right)^\intercal \boldsymbol{n}_{s,\tau}^k\right| < \varepsilon_n, \tag{9}$$

where ε_n represents the maximum angle deviation between normals in the scene and the model.

Each geometrically consistent correspondence cluster is used to estimate a transformation, aligning a specific model with the scene under consideration. Because not all correspondences within a geometrically consistent cluster are representatives of valid rigid transformations, each cluster is post-processed by a RANSAC stage in order to eliminate outliers prior to the pose estimation of the object.

F. Hypotheses Verification

Each constructed hypothesis is individually registered to the point cloud of the scene by ICP and verified by the method proposed in [17] extended to use color information as described in [9]. In order to account for dynamic changes, please note that in contrast to [9], hypotheses in this work are verified against the point cloud obtained from the current vantage point only. While past correspondences are only kept for T time steps, correspondences that form a cluster for a verified hypothesis are kept longer than that, until the respective verified hypothesis disappears as a whole for another T time steps. The reason is that we do not want to loose good correspondences that for instance have moved to the backside of an object as the camera has moved around that object.

IV. RESULTS

We evaluated the system on the TUW [1] and the Willow [2] RGB-D dataset. The TUW dataset contains 15 sequences with highly occluded and cluttered tabletop objects annotated in static environments, where multiple object instances are present in some of the views of the dataset. The TUW model database consists of $|\Omega| = 17$ models with a maximum extent of 30 cm, which are partly symmetric and some lack distinctive surface texture. To measure the performance of the proposed method in sequences with temporal changes of the objects, in position and presence in the scene, we extended the (static) TUW dataset by observations of two table tops and one kitchenette, which have been captured over an extended period of time by a Kinect camera mounted on the STRANDS robot Werner[3]. All view points in the TUW dataset were chosen manually by the user such that there is some overlap between successive views within a sequence. Using the semi-automated ground-truth annotation

[1] goo.gl/qXkBOU
[2] http://rll.berkeley.edu/2013_IROS_ODP/
[3] http://strands.acin.tuwien.ac.at/

tool [9], these *dynamic* environments were annotated and uploaded to the TUW dataset website. In the following, these (dynamic) sequences are referred by $\psi \in \{16, 17, 18\}$.

A. Test setup

We tested our method against the single-view only recognition system proposed by Aldoma *et al* [4] and the multi-view method proposed in our previous work [10].

The input point clouds of the test sequences are processed in alphabetically order of the given view names for all systems, which produce intermediate results after each timestep. The correspondences for each input cloud are obtained by the same single-view recognition pipeline [4] using keypoints extracted by DoG and matched to model keypoints by first nearest neighbor search with respect to their SIFT and SHOT description. The multi-view method [10] creates hypotheses by clustering geometrical consistent correspondences from the same view, accumulating them into a common reference frame and verifying them against a 3D reconstruction of the scene. In contrast, the proposed feature integration method clusters correspondences merged from multiple views (described in Section III) and verifies them against the current camera viewpoint only. Both multi-view methods accumulate information from up to $T = 10$ most recent views of the scene. Clusters for all methods are created by at least 7 (Willow: 5) correspondences with the constraints $\varepsilon = 15$ mm and $\varepsilon_n = 0.2$. Each pose estimate of the generated hypotheses is refined by 10 iterations of ICP registering the individual objects to the point cloud of the scene. The final hypotheses verification stage uses color variance parameters for the normalized LAB color space of $\sigma_L = 0.6$ and $\sigma_{AB} = 0.5$ (Willow: $\sigma_L = 0.1$, $\sigma_{AB} = 0.1$), an inlier threshold of 15 mm and a resolution of 5 mm. These parameters were empirically evaluated on the first four sequences of each dataset.

The performance is measured by precision and recall, where a true positive is counted if a detected object is within 3 cm of a ground-truth object of the same class with respect to its centroid. To neglect ground-truth objects outside the field of view or invisible for the current camera orientation, these values were only computed for instances with a ground-truth occlusion (see [9]) $\leq 95\%$. We illustrate the results by averaging all intermediate precision and recall values for the tested sequences (Table I), by showing the improvement of performance over the number of observations taken into account by the system (Fig. 4), by measuring the mean error in translation and rotation as well as calculating the total precision and recall values (Table II).

B. Evaluation on a Static Environment

This subsection presents the evaluation on the *static* environments in the TUW and Willow dataset. Overall, Table II shows that the single-view method achieves slightly higher precision compared to the tested multi-view methods, which can be explained by the fact that all correspondences generating an object hypothesis are coming from the current view of the scene. Neglecting sensor noise of the camera, these keypoints are usually quite robust as they are observed at the same time (e.g. avoiding changing lighting conditions) and do not have to be transformed into another coordinate system by a potentially noisy camera pose estimate. As the reason for objects not being detected by the single-view recognizer often is a too low number of keypoint matches, any cluster of correspondences generating these hypotheses after being transformed by our approach is more likely to be small as well (and therefore generating weaker hypotheses and slightly worse precision rates) compared to hypotheses generated by clusters from single-view processing only.

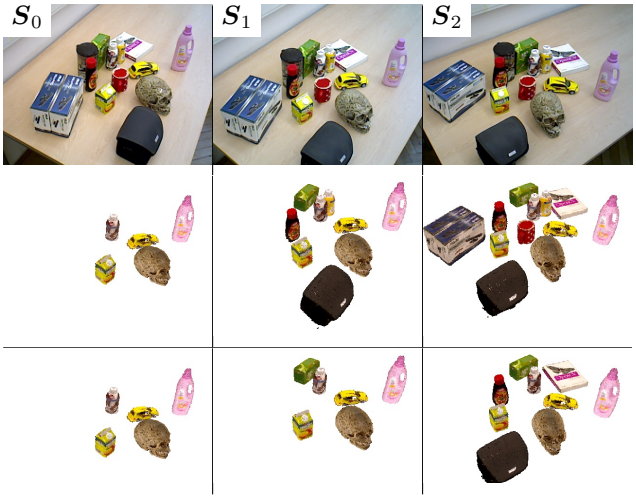


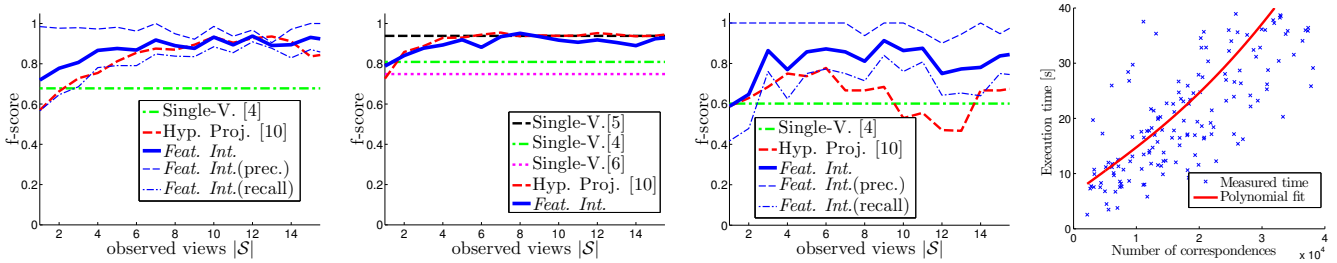
Fig. 5: Example results for sequence $\psi = 6$ of the TUW dataset. **Top:** Scene S_t observed by the robot, **Middle:** Recognized objects by the proposed multi-view method, **Bottom:** Recognized objects by the single-view system. While the result for the first view $\mathcal{S}_0$ is the same (no prior information), the next frame recognizes three additional objects. After three observations, the proposed method is already able to correctly recognize all objects in the scene but two, the coffee container and the lower green tea box.

Fig. 4a and 4b show the f-score with respect to the number of observed views in the static TUW and Willow dataset, respectively. Even very few observations of the environment lead to a superior f-score rate, which is particularly due to the increased recall gained by merging hypotheses or feature correspondences. Since the completely static environments can be correctly reconstructed by accumulating scene point clouds over time, the proposed feature integration method gains only slightly better recognition rate compared to our previous multi-view approach [10].

As an example, Fig. 5 shows the recognition result for the first three views of a sequence in the TUW dataset processed by the single- and proposed multi-view approach. From this example, it can be concluded that the low-level information stored by the multi-view system particularly enhances recognition rates for objects that are cluttered by other objects or lack distinctive texture.

		static TUW dataset															*dynamic*		
ψ		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
SV [4]	precis.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
[10]	precis.	1.00	1.00	0.97	0.95	1.00	1.00	0.89	1.00	1.00	0.91	1.00	1.00	0.98	1.00	1.00	0.95	0.92	0.75
FeatInt	precis.	0.89	0.98	0.94	0.99	0.97	0.99	0. 98	0.93	0.86	1.00	0.96	0.97	0.89	0.99	0.97	0.97	0.98	1.00
SV [4]	recall	0.43	0.63	0.52	0.58	0.43	0.72	0.94	0.62	0.00	0.17	0.81	0.74	0.37	0.60	0.35	0.43	0.37	0.60
[10]	recall	0.58	0.63	0.85	0.74	0.70	0.84	0.88	0.80	0.00	0.63	0.81	0.89	0.57	0.71	0.52	0.56	0.39	0.45
FeatInt	recall	0.62	0.75	0.77	0.78	0.80	0.96	1.00	0.98	0.13	0.50	0.92	0.89	0.70	0.79	0.56	0.75	0.71	0.59
#views		7	7	14	16	10	13	8	8	9	7	9	8	18	16	13	21	21	19
#objects		53	62	78	258	186	180	48	40	45	16	53	66	82	198	122	201	227	92

TABLE I: Precision (top) and recall (middle) for the 15 *static* environments of the TUW dataset and the three *dynamic* environments. The values are evaluated for the single-view system (SV), the multi-view hypotheses projection method [10] and the proposed approach (*FeatInt*). The total number of ground-truth objects visible (occlusion $\leq 95\%$) in each sequence is shown in the bottom row.



(a) TUW static (b) Willow (c) TUW dynamic (d) Computation time

Fig. 4: Average f-score for the systems single-view (green dashed dotted), the multi-view method [10] (red dashed) and our proposed approach (blue solid) evaluated on the 15 *static* environments of the TUW dataset (left), the 24 *static* sequences of the Willow dataset (second left) as well as the three *dynamic* TUW environments (second right). The average is taken over all scene observations at time t. Our proposed multi-view method accumulates correspondences from up to $T = 10$ observations, which results in a computation time behaviour as shown on the right.

C. Evaluation on Static Environments with Dynamic Objects

To evaluate the performance for environments not static over the whole observation period, this subsection evaluates on observations, where objects have been moved to other locations or (dis-)appear from the field of view. Fig. 4c and the bottom part of Table I show the results on these dynamic sequences. While the difference between the proposed method and the single-view system is similar to the evaluation results for the static environments, the decrease in performance of the hypotheses projection method [10] becomes very evident for these dynamic scenarios. Due to the broken assumption of a static environment, this method verifies transferred hypotheses generated in the past against an accumulated point cloud with all scene points ever observed. This leads to a high number of false positives and makes it infeasible in environments that cannot be considered static.

D. Computation Time

This subsection evaluates the computation time of the methods on an Intel Quad Corei7 (2.8GHz) system with 8GB RAM, which is shown in Table III for the tested multi-view systems. To estimate the scale of the computational complexity by the number of correspondences stored by the multi-view system, we fitted a 4th degree polynomial to 221 measurements of the execution time (Fig. 4d). Considering the coefficients of the polynomial ($p_1 \ll p_2 \ll p_3$), we can assume an approximate linear increase in computation time with respect to the number of correspondences $\left|\mathcal{C}_{\omega,t^+}\right|$, scaled by the first-order term p_1. This dependency, although not fulfilling the worst-case assumption of $\mathcal{O}\left(\left|\mathcal{C}_{\omega,t^+}\right|\right)$, still shows the importance of keeping the amount of correspondences low by removing redundant information as described in Sections III-B and III-D.

		μ_r ($\pm\sigma_r$) [deg]	μ_t ($\pm\sigma_t$) [mm]	total prec.	total recall
Single-V. [4]	*Willow*	4.8 ($\pm$3.6)	7.3 ($\pm$4.7)	0.94	0.71
Hyp. proj. [10]	*Willow*	3.9 ($\pm$3.2)	6.1 ($\pm$3.9)	0.94	0.90
Feat. Int.	*Willow*	5.0 ($\pm$3.8)	7.5 ($\pm$4.7)	0.92	0.89
Single-V. [4]	*TUW st.*	4.0 ($\pm$2.9)	6.8 ($\pm$4.1)	0.99	0.62
Hyp. proj. [10]	*TUW st.*	3.6 ($\pm$3.4)	6.1 ($\pm$7.4)	0.96	0.72
Feat. Int.	*TUW st.*	4.4 ($\pm$3.6)	6.7 ($\pm$3.9)	0.97	0.78
Single-V. [4]	*dynamic*	4.8 ($\pm$3.4)	6.6 ($\pm$4.2)	1.00	0.43
Hyp. proj. [10]	*dynamic*	6.6 ($\pm$4.9)	7.7 ($\pm$5.3)	0.90	0.47
Feat. Int.	*dynamic*	6.0 ($\pm$4.6)	8.1 ($\pm$5.3)	0.98	0.70

TABLE II: Average error for rotation μ_r (with standard deviation σ_r) and translation μ_t (with standard deviation σ_t) for all true positive objects within the Willow (top), the static TUW (middle) and the recorded dynamic TUW dataset (bottom). A recognized object is counted as true positive if the translational error is $\leq$ 3 cm and the rotational error $\leq 30^\circ$ with respect to the corresponding ground truth object.

Number of observations	1	2	3	4	5	6
Hyp. proj. [10]	3.1	5.8	8.7	12	15.9	19.2
Feat. Int.	8.9	12.8	16.5	23.2	27.4	34.8

TABLE III: Average time in seconds to compute recognition results given a certain number of observed views.

V. CONCLUSIONS

We have shown a recognition system that accumulates keypoint correspondences over multiple views and transfers this low-level information into the current recognition problem to achieve improved recognition results in both, static and partially dynamic environments. The evaluation on challenging scenes, highly cluttered and containing multiple occluded objects, showed that our proposed method achieves state-of-the-art results on static environments. For dynamic environments with objects moving to other locations, we could show significant improvements in terms of precision and recall. Making fewer assumptions on the recognition task, we believe that this enables our method to be deployed on a wide range of robotic systems.

A limitation of the system is the approximate proportional increase of computational complexity with the size of stored keypoint correspondences, making it infeasible for many real-time applications. Although the method has been partially implemented for parallel processing, the work so far concentrated on improving recognition results. A significant speed up could probably be achieved by further parallelization, particularly using GPU programming. Furthermore, to avoid re-computation of correspondences already clustered during previous observations of the scene, the multi-view system could be extended by additionally caching cluster information, which is potential future work.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS.

REFERENCES

[1] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *The International Journal of Robotics Research*, 2012.

[2] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1284–1306, 2011.

[3] 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 *Asian Conference on Computer Vision (ACCV)*. Springer, 2013, vol. 7724, pp. 548–562.

[4] A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. Di Stefano, and M. Vincze, "Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DoF pose estimation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 2104–2111.

[5] Z. Xie, A. Singh, J. Uang, K. S. Narayan, and P. Abbeel, "Multimodal blending for high-accuracy instance recognition," in *Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013.

[6] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2012.

[7] L. Kunze, C. Burbridge, M. Alberti, A. Thippur, J. Folkesson, P. Jensfelt, and N. Hawes, "Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding," in *Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014.

[8] T. Krajník, J. Pulido Fentanes, G. Cielniak, C. Dondrup, T. Duckett, *et al.*, "Spectral analysis for long-term robotic mapping," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[9] A. Aldoma, T. Fäulhammer, and M. Vincze, "Automation of ground truth annotation for multi-view rgb-d object instance recognition datasets," in *Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014.

[10] T. Fäulhammer, M. Zillich, and M. Vincze, "Multi-view hypotheses transfer for enhanced object recognition in clutter," in *IAPR Conference on Machine Vision Applications (MVA)*, 2015.

[11] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 48–55.

[12] I. Gordon and D. G. Lowe, "What and where: 3d object recognition with accurate pose," in *Toward category-level object recognition*. Springer, 2006, pp. 67–82.

[13] S. Ekvall, D. Kragic, and F. Hoffmann, "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling," *Image and Vision Computing*, vol. 23, no. 11, pp. 943–955, 2005.

[14] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2050–2055.

[15] R. Pless, "Using many cameras as one," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–587.

[16] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2006, pp. 3760–3767.

[17] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification method for 3d object recognition," in *European Conference on Computer Vision (ECCV)*, 2012.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[19] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of Histograms for local surface description," in *European Conference on Computer Vision (ECCV)*, 2010.

[20] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3d reconstruction and tracking." in *3DIMPVT*. IEEE, 2012, pp. 524–530.

[21] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP*. INSTICC Press, 2009.

[22] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 80–91, 2012.

[23] F. Tombari and L. Di Stefano, "Hough voting for 3d object recognition under occlusion and clutter," *IPSJ Trans. on Computer Vision and Applications (CVA)*, vol. 4, pp. 20–29, 2012.