A Multi-Modal RGB-D Object Recognizer

Thomas Fäulhammer[1] Michael Zillich[1] Johann Prankl[1] Markus Vincze[1]

Abstract— **In this paper we propose a multi-modal object recognition system that uses a two-step hypothesis verification approach to improve runtime efficiency. The system uses local and global appearance and shape features, generating many possibly competing hypotheses, which are then verified such that the scene can be optimally explained in terms of recognized object models. The introduced modification in this time consuming step reduces runtime considerably, while maintaining recognition performance. We evaluate recognition performance for various feature extraction modalities on the publicly available Willow Garage RGB-D dataset and show runtime improvements of a factor 2 to 10.**

I. INTRODUCTION

Recognition of object instances together with pose recovery is a key step in various robotics and computer vision applications such as manipulation, surveillance or augmented reality. Some recent approaches showed significant performance improvements when using multi-modal cues [1], [2], [3] which often comes at the cost of increased computational load. In this work we propose a multi-modal object recognition approach that reduces computational complexity in the most time consuming step of globally verifying object hypotheses, thus reducing runtime while maintaining recognition performance. Similar to [4], we use multiple parallel recognition pipelines which can be categorized into two types: a pipeline matching features extracted *local*ly to detect objects in presence of clutter and occlusion, and a pipeline with a richer global description for clear objects which are easy to segment from their immediate environment.

The main contributions of this work are

- A novel two step approach for hypotheses verification that reduces the computational complexity of a global only approach such as in [1].
- integration of 6DoF pose recovery from global description by Ensemble of Shape and CNN based features
- addition of a pose clustering stage for improved computational efficiency
- an evaluation of the different recognition pipelines in terms of computation time and recognition rate under various occlusion levels on the Willow RGB-D dataset
- a publicly available recognition framework[1]- optimized for memory efficiency and multi-core computers.

II. RELATED WORK

There is a vast literature on 3D object recognition [5] but only few are able to cope with clutter and multiple object instances present in the scene.

Tang *et al.* [6] describes each object by a hue histogram and a set of SIFT features with corresponding 3D keypoint locations extracted from rendered views of the object. At test time, the input point cloud gets segmented by finding Euclidean clusters above a table plane. Each segment generates multiple object hypotheses by a Naive Bayes Nearest Neighbor approach with respect to its hue histogram and bag-of-sift features. To estimate the object's pose, a RANSAC based approach tries to minimize the re-projection error of SIFT correspondences. This however adds other challenges such as clustering feature correspondences to multiple object instances present in the scene which we solve by ensuring geometric consistency for these correspondences [1]. Another major difference is the hypotheses verification. While [6] verifies object hypotheses individually by checking the amount of explained keypoints, we project the trained 3D object model in the scene and perform multi-modal (i.e. color and 3D distance) checks for each visible point.

Xie *et al.* [2] showed that dense SIFT matching improved the recognition results significantly compared to sparse keypoint extraction. They propose a multimodal blending approach for hypothesis verification using SIFT, shape and color models. While they achieved excellent results for the Willow dataset, their approach assumes textured objects and test objects standing on a tabletop and being easily to segment. We propose a more general solution to get rid of these assumptions.

Using simulated 2.5 views from 3D object models, Bonde *et al.* trains a soft-label random forest with features encoding depth edge orientations, occlusion and pose of the object. Depth discontinuities in the range image are re-projected into a voxel grid and accumulated into a histogram to find dominant edge orientations. Occlusion is simulated by placing occluders between the simulated camera and the voxelized object. The object pose is encoded by quantizing the simulated pose into 16 pose classes. Using a margin-maximization training scheme on around 50000 simulated training views for each object, this learning approach jointly classifies location and pose of an object instance even in presence of occlusion and clutter.

Papazov *et al.* [7] proposed a RANSAC based object recognition approach which verifies object hypotheses by an acceptance function. Object hypotheses are generated by sampling scene point pairs at a certain Euclidean distance to each other, encoding their relative position and surface orientation and finding corresponding model point pairs efficiently by the use of a hash table. The acceptance function is computed on all model points projected into the scene with

[1]All authors are with the Vision4Robotics group, Automation and Control Institute, Vienna University of Technology, Austria faeulhammer, zillich, prankl, vincze@acin.tuwien.ac.at

[1]https://github.com/strands-project/v4r

the estimated transformation and consists of a support and a penalty term. The support and penalty terms are proportional to the number of model points which are close to or occlude any scene point, respectively. A final filtering step checks for conflicting hypotheses by creating an octree containing all visible model points in it. If a leaf node is occupied by points from more than one object hypothesis there is a penalty. Apart from the significant difference in the generation of object hypotheses, our method also differs in the verification stage. The main differences are that we also check for color similarity and compute conflicts based on the overlap of the rendered images of hypotheses.

Our object recognition framework is based on the work of Aldoma *et al.* [1]. It uses multiple pipelines; each pipeline extracting a different type of feature description either providing *local* feature correspondences or matches of segmented clusters to object models in certain poses by using a *global* OurCVFH [8] description. After merging the generated hypotheses, a global verification approach finds the subset of object models that best explain the scene in terms of shape, color and clutter.

III. METHOD

The task is to detect pre-trained objects in a scene and estimate their 6 degree of freedom (DoF) pose with respect to the camera. The scene $\mathcal{S}$ is sensed by an RGB-D camera which provides for each pixel of the image both color and depth information.

A. Training object models

We train objects in a controlled setup (turntable) where we record a number of RGB-D training views with associated camera pose. The camera pose is tracked using the method of Prankl *et al.* [9]. From each training view, we extract a set of feature descriptors from the segmented object. As in [1], a multi-pipeline, shown in Fig. 1 extracts local as well as global features. In our work, we decided to use SIFT [10] and SHOT [11] for local, and ESF [12] and a CNN based approach [13] for global description. Keypoints for SIFT are sampled by DoG on the 2D image of each training view and stored together with the corresponding depth value. Keypoints for SHOT are sampled uniformly across each training view. While SIFT proved to be reliable for textured objects, the descriptiveness of SHOT deteriorates with noise [14] (e.g. distant objects sensed by a Kinect sensor). To reduce the amount of noisy and indistinct features, we skip SHOT feature descriptions from keypoints further away than a distance z^t_{SHOT} from the camera, and also for keypoints at planar patches (i.e. where the surface curvature is smaller than a threshold).

Additionally, we compute 3D models $\mathcal{M}$ of each object. These 3D models are used in the final verification stage to synthetically generate a scene from recognized objects and find the subset that best fits the recorded scene. The 3D models are created by merging the segmented point clouds in each training view into a common coordinate system. To take into account the camera noise model [15] and reduce the memory footprint, points close to depth discontinuities are removed and the remaining points put in an octree (resolution set to 1mm). Within each node of the octree, only the points with the lowest noise level are kept to form $\mathcal{M}$. The output of the training stage is a set of object models

B. Generating object hypotheses

To generate object hypotheses, we follow a similar approach as Aldoma *et al.* [1].

a) Local pipeline: Given an RGB-D image of the scene, we extract the same set of features as described in Section III-A. Using fast approximate nearest neighbor search [16], each feature is matched to its k nearest features from $\mathcal{M}$ with respect to their L2 distance. To detect a single object instance in a scene, a RANSAC based transformation estimation would directly generate an object hypothesis. However, to allow detecting multiple instances of the same object in a scene, [1] stores keypoints associated to the features in a graph. Each node in the graph represents a keypoint match and gets connected to other nodes if they belong to the same object model and the associated keypoints are geometrically consistent. Two keypoint correspondences are geometrically consistent if Euclidean distance as well as the relative surface normals of scene and model keypoints conform. Aldoma *et al.* [1] search this undirected graph for maximal cliques using the depth-first search algorithm in [17]. For cliques with a clique number (amount of geometrically consistent keypoint correspondences) $\geq c^t$, an object hypothesis is generated with a pose $\mathcal{T}$ estimated using RANSAC.[2]

b) Global pipeline: To find objects using global descriptors, we first segment $\boldsymbol{S}$ into a set of point cloud clusters. As the performance of the global pipeline strongly depends on the outcome of the segmentation method, it is important to choose a segmentation method that suits the task at hand. In our method, we search for a dominant plane using RANSAC and segment points on top by Euclidean clustering. Each cluster $\boldsymbol{C}_i \in \boldsymbol{S}$ is then described with respect to its shape (ESF) and visual appearance. The visual appearance is encoded by feeding the segmented and cropped RGB image into the Convolutional Neural Network (CNN) proposed in [13] and pre-trained on the ILSVRC-2012 competition with 10 million images of 10.000+ different categories [18]. Rather than fine-tuning this network on our set of objects, we decided to use the 4096 dimensional feature vector extracted from the last layer of the network and train a multi-class linear SVM [19] with all our segmented training views. While ESF features are matched using nearest neighbor search to its k_{ESF} closest training views, we use the k_{CNN} most likely objects returned by the SVM.

As both of these feature descriptors do not encode the scale of the object, we add an additional size constraint. In particular, we reject clusters if their dimensions along the two prominent Eigenvectors is $\tau_{\text{c,max}}$ larger or $\tau_{\text{c,min}}$ smaller than the ones measured on the 3D point cloud model of

[2] At least 3 keypoint correspondences are required to estimate a 6dof pose.

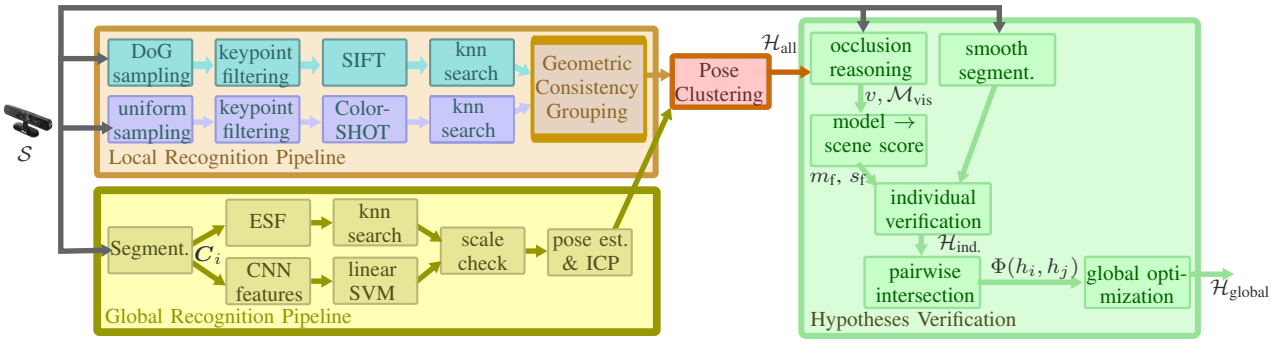


Fig. 1: Workflow of the proposed recognition system. The scene $\mathcal{S}$ is described by multiple local and global features accounting for shape and visual appearance properties. The extracted features are matched to pre-trained features of the object models to generate object hypotheses $\mathcal{H}$. The hypotheses are clustered and verified with respect to geometry and color in a two-step approach; a fast *individual* and a *global* verification step. The output is the set of recognized objects $\mathcal{H}_{\text{global}}$.

the matched object. To recover the 6DoF object pose, we align the centroid of the object with the centroid of $\boldsymbol{C}_i$ downprojected onto the dominant table plane.[3] Next, we align the z axis of the object with the normal vector of the dominant plane and span the x-y axes arbitrary onto the table plane (i.e. orthonormal to z). The remaining degree of freedom, i.e. the orientation around z, is sampled uniformly with a step size of $\delta_\gamma[^\circ]$. Each of these $\left\lceil \frac{360}{\delta_\gamma[^\circ]} \right\rceil$ $k_{\text{ESF/CNN}}$ possible object poses is afterwards refined using ICP. Note that we decided to keep all refined object poses (and not only the one with the best ICP fitness score) to cope with geometrically symmetric objects.

C. Pose clustering

The output of previous stage is a set of object hypotheses

$$\mathcal{H}_{\text{all}} = \{m_i, \mathcal{T}_i : m_i \in \mathcal{M},\, \mathcal{T}_i \in \mathbb{R}_{4\times4}\}\,, \tag{1}$$

where each hypothesis contains the matched object model m and its estimated object pose $\mathcal{T} = [\boldsymbol{R}|\boldsymbol{t}]$. The object pose hereby represents the rigid body transformation aligning the object to the camera coordinate system by a rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$. As the hypotheses set potentially contains redundant elements leading to increased computational complexity, we cluster nearby hypotheses based on their position and orientation. In particular, starting with a random seed object hypothesis $h^j \in \mathcal{H}_{\text{all}}$ we iteratively cluster hypotheses $h^i \in \mathcal{H}_{\text{all}}$ *iff*

$$\begin{aligned} &m^j = m^i \quad \text{and} \quad \|\boldsymbol{t}^j - \boldsymbol{t}^i\| < \delta_t \quad \text{and} \\ &r^{j\to i}(\alpha), r^{j\to i}(\beta), r^{j\to i}(\gamma) < \delta_r, \end{aligned} \tag{2}$$

where $\boldsymbol{R}^{j\to i} = \boldsymbol{R}^j \boldsymbol{R}^{i^T}$ gives the relative orientation of the two hypotheses and $r(\alpha)$, $r(\beta)$ and $r(\gamma)$ are yaw, pitch and roll angles of rotation matrix $\boldsymbol{R}$. The threshold parameters δ_t and δ_r define the maximum allowed relative distance and orientation for two object hypotheses to be clustered together; they influence computation time and accuracy. The clustering is repeated till all hypotheses are assigned to a cluster of size ≥ 1. The output is a reduced set of hypotheses $\mathcal{H}$.

[3]Note that [9] aligns the object models s.t. the centroid defines the origin of the model coordinate system and the z axis is parallel to the normal plane of the turntable. Our pose recovery method of the global pipeline implicitly assumes that these object models are observed in the same upright position.

D. Hypotheses verification

The goal of the verification stage is to reject falsely generated object hypotheses by comparing them with respect to the given input cloud. We use a two-step approach for rejection of object hypotheses. First, we reject potentially wrong hypothesis *individually* by occlusion reasoning and computing a model fitness for each $h \in \mathcal{H}$. Second, we *globally* optimize a cost function to decide which of the remaining hypotheses best explain the scene.

1) Individual Verification: The goal of the individual verification step is to reject hypotheses which either contradict with the observed scene S by occluding parts of it or weakly explaining it in terms of geometry and/or color. Rejection of these hypotheses allows us to reduce the computational complexity in the following global verification stage.

a) Occlusion Reasoning: We first check which part of the 3D object model is visible from the current viewpoint. To infer the visible model points $\mathcal{M}_{\text{vis}} \subset \mathcal{M}$, we compute self-occlusion and the occlusion from observed scene points by projecting each object hypotheses into the camera coordinate system using the estimated object pose $\mathcal{T}$. Using the camera intrinsics and doing depth buffering onto the image plane, we define a visibility ratio v for each hypothesis h_i by $v_i = \frac{|\mathcal{M}_{i,\text{vis}}|}{|\mathcal{M}_i|}$.

b) Model / Scene Fitness: Using the visible model cloud $\mathcal{M}_{\text{vis}}$, we check how well it fits $\mathcal{S}$ in terms of geometry and color. Given a point $\boldsymbol{p}_m \in \mathcal{M}_{\text{vis}}$, we search for nearby scene points $\mathcal{N}(\boldsymbol{p}_m) = \{\boldsymbol{p}_s \in \mathcal{S} : \|\boldsymbol{p}_m - \boldsymbol{p}_s\| < \rho\}$ and define a model fitness term

$$m_f(\boldsymbol{p}_m) = \min_{\boldsymbol{p}_s \in \mathcal{N}(\boldsymbol{p}_m)} \exp\left(-d(\boldsymbol{p}_m, \boldsymbol{p}_s)\right) \tag{3}$$

with

$$d(\boldsymbol{p},\boldsymbol{q}) = \frac{\|\boldsymbol{p}-\boldsymbol{q}\|^2}{\sigma_{3D}^2} + \frac{\|c_L(\boldsymbol{p}) - c_L(\boldsymbol{q})\|^2}{\sigma_L^2} + \frac{\|c_{AB}(\boldsymbol{p}) - c_{AB}(\boldsymbol{q})\|^2}{\sigma_{AB}^2} + \frac{\arccos\left(\boldsymbol{n}(\boldsymbol{p})^T \boldsymbol{n}(\boldsymbol{q})\right)}{\sigma_n}, \quad (4)$$

where $c_L(\boldsymbol{p})$ are the L and $c_{AB}(\boldsymbol{p})$ the A and B components of the point color of $\boldsymbol{p}$ in the CIELAB color space. The vector $\boldsymbol{n}(\boldsymbol{p})$ represents the surface normal at $\boldsymbol{p}$. The variables σ_{3D}, σ_L, σ_{AB} and σ_n are the corresponding scale factors and influence the importance of each term. The fitness term ranges between 1 for a perfect fit and 0 for model points that can not be explained by any points in $\mathcal{S}$. The average model fitness for a hypothesis is defined by

$$\overline{m_f} = \frac{\sum_{p \in \mathcal{M}_{vis}} m_f(\boldsymbol{p})}{|\mathcal{M}_{vis}|}. \quad (5)$$

and defines together with a threshold parameter m_f^t if the hypothesis gets rejected. Instead of using a hard threshold, we scale m_f^t linearly with the occlusion ratio $(1 - v)$. Intuitively, the less we see from an object, the more difficult it is to judge if the hypothesis is correct. Therefore, we require a better fit to reduce the influence of noisy measurements.

c) Smooth segmentation: As shown by our experiments, in many cases local shape features within our model database erroneously fit parts of the scene potentially generating wrong hypotheses. As our optimization strategy aims to maximize the number of explained scene points, these hypotheses would be accepted if the model fit is sufficiently large (e.g. flat objects like books locally have the same shape distribution as the edge of a table). To penalize these circumstances, we extract smooth surface patches from $\mathcal{S}$ and reject objects whose visible model points only partially fit a patch. The surface patches are extracted by selecting random seed points in S and iteratively clustering nearby points with similar surface normals and curvatures.

2) Global Verification: The output of the individual verification is a set $\mathcal{H}_{ind.} \subset \mathcal{H}$ containing all hypotheses for which $\overline{m_f} > m_f^t$. As a final step in our pipeline, we *globally* search for the best fitting set of hypotheses $\mathcal{H}_{global}$. As we already removed weak hypotheses in the previous stage, the final step penalizes conflicting hypotheses. To penalize conflicting hypotheses, each hypothesis is back-projected onto the image plane and for each pair $h_i, h_j \in \mathcal{H}_{ind.}$ we compute an intersection penalty $\Phi(h_i, h_j)$ as the ratio of overlapping to total number of occupied pixels. The goal of the optimization function is then to minimize the overall intersection penalty while at the same time maximizing model and scene fitness. The scene fitness $s_f(\boldsymbol{s})$ for a point $\boldsymbol{s} \in \mathcal{S}$ is hereby defined by the best matching visible model point

$$s_f(\boldsymbol{s}) = \min_{\boldsymbol{p} \in \mathcal{M}_{vis}} (m_f(\boldsymbol{p}) : \boldsymbol{s} \in \mathcal{N}(\boldsymbol{p})). \quad (6)$$

As in [1], we define the problem as a combinatorial problem where we optimize over a boolean vector $\boldsymbol{x} \in \{0,1\}^{|\mathcal{H}_{ind.}|}$. Each element in $x_i \in \boldsymbol{x}$ represents a hypothesis $h_i \in \mathcal{H}_{ind.}$ and is set to 0 or 1 depending on rejection or acceptance, respectively. To get the final set of hypotheses, we solve following optimization problem

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \frac{\lambda_m}{|\mathcal{H}_{ind.}|} \sum_{h_i \in \mathcal{H}_{ind.}} m_f(h_i)\, x_i + \frac{\lambda_s}{|\mathcal{S}|} \sum_{\boldsymbol{p} \in \mathcal{S}} s_f(\boldsymbol{p}) - \lambda_\Phi \sum_{h_i \in \mathcal{H}_{ind.}} \sum_{h_j \in \mathcal{H}_{ind.}} \Phi(h_i, h_j)\, x_i x_j \quad (7)$$

$$\text{s.t. } x = \{0,1\} \in \mathcal{X},$$

with regularization parameters λ_m, λ_s and λ_Φ.

The cost function in Eq. 7 is optimized using METSlib [20] from an initial solution with all hypotheses disabled and using local search over all neighboring solutions with a Hamming distance not greater than 2. The output of an example test scene is shown in Fig 2.

IV. RESULTS

To evaluate the performance of our system, we test it on the Willow RGB-D dataset used for the ICRA perception challenge 2011(`goo.gl/qXkBOU`). The training set of these datasets consist of 33 rigid textured object models, each object recorded from 37 different training views (corresponding to a 10 degree viewpoint change). The test set consists of 24 sequences of recorded table top scenes from different viewpoints with multiple objects present in each scene. There is a total of 353 test views with 3257 object occurrences. The ground-truth of each object occurrence is represented as the 6 DoF pose aligning the object model with the scene and an occlusion value which is equal to the ratio of model points visible in the respective test view. We measure f-score $f = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ for various feature selections and compare it to the results in [1], [6] and [2]. Additionally, we evaluate how well objects are detected under different levels of occlusion. The parameters for the following evaluation were validated empirically on 10% of the test set. As we were mainly interested in the relative performance of our tested pipelines to each other, we spared extensive fine-tuning of the parameters and kept them at the same values over all evaluations. The chosen values are listed in Table I.

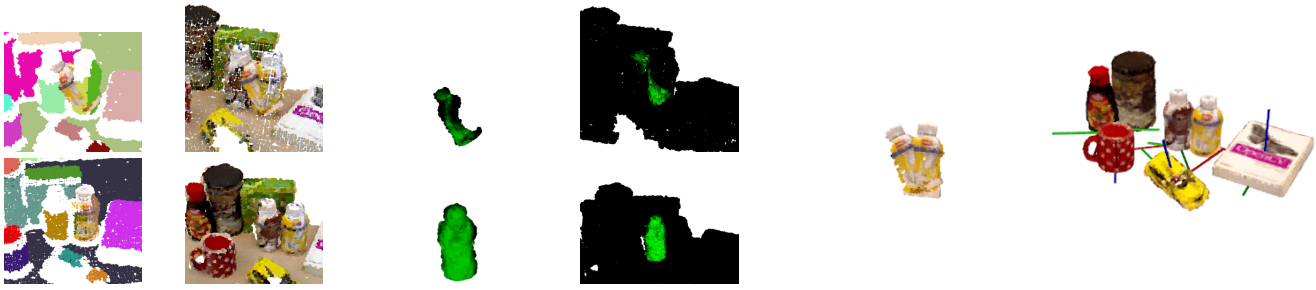
A. Feature Evaluation

In this test, we enabled subsets of pipelines shown in Figure 1 and measured their performance. As shown in Table II, we achieve baseline performance for all tested combinations using SIFT descriptors. The best trade-off between f-score and computation time was achieved by the local pipeline combination SIFT + SHOT. Comparing recall, it shows that considerably fewer objects are missed by using multiple pipelines describing multiple modalities over both local and global regions. The decreased precision for recognition pipelines using ESF can be explained by the fact that some objects in the Willow dataset are geometrically very similar (e.g. cereal box, bottles) and so generate many false hypotheses that need to be rejected by the verification stage. Hypotheses of similar shaped objects can however only be distinguished by the color term in Eq. 3 in our approach. Even though using the CIELAB color

Test scene $\mathcal{S}$

top: Local pipeline with uniformly (green) and filtered non-planar(red) keypoints (left); local feature correspondences for an object (middle); and geometric consistent hypotheses (right)
bottom: global pipeline with segmentation output (left), generated hypotheses by centroid alignment and sampling uniformly around the z axis (middle); and filtered hypotheses by scale check (right)

merged hypotheses



smooth seg. | model scene overlay | model fitness (green) | scene fitness (green) | intersection Φ | verified hypotheses $\mathcal{H}_{\text{global}}$

Individual verification for an object hypotheses in wrong (**top**) and correct (**bottom**) pose

Fig. 2: Example output of various components within the proposed recognition system. Best viewed in color.

space, we experienced a considerable change of the point color under different lighting conditions and so decided to use conservative weight factors. Other than increasing these weight factors, another remedy to reject these hypotheses could be that the verification stage also takes into account the probability distributions from the feature matching output.

We also evaluated how well objects are recognized at different degrees of occlusion by various types of feature extractors in our system. As shown in Figure 3 SIFT accounts for most of the recognition performance; especially under heavy occlusion where global pipelines either fail to segment the object or suffer from the few points being visible for a meaningful global feature description. However, for less occluded object, using multiple of the proposed recognition pipelines increases recognition rate by up to about 20%.

B. Computation time

We ran our evaluations on an Intel 4-core i7 CPU with 32GB RAM. While the computation time for [4] has been measured on the same machine, the computation times for [6] and [2] have been taken from the respective papers both of which using a comparable system. Although there is no computation time given for [1], we expect it to be close to [4] due to the similarities of these systems. Table II shows the approximately 2×speed-up we gain from using pose clustering and our proposed combination of individual and global verification compared to the fastest tested state-of-the-art approach when disabling global pipelines. The runtime increases significantly when enabling any global pipelines. This can be explained by the large number of object hypotheses that need to be verified which depends on the number of clusters being extracted by the segmentation algorithm and the amount of object hypotheses being generated, i.e. the sampling ratio of the angle around the z-axis. This highlights the importance of our proposed scale check filter and pose clustering stage.

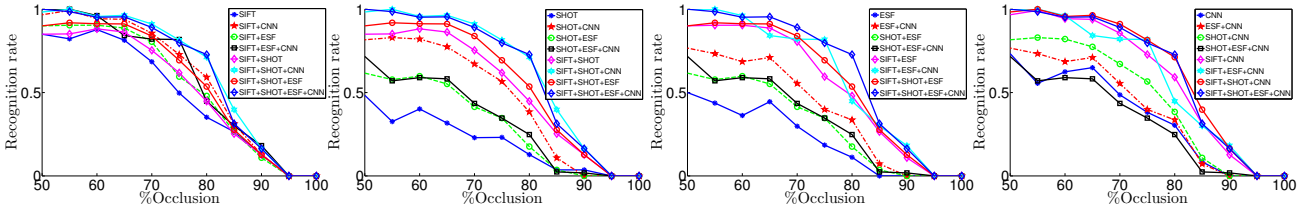
V. CONCLUSIONS

We have presented a recognition framework extracting multiple modalities from local and global parts of the objects and fusing them into a compact set of object hypotheses. Our generic pipeline allows us to use various types of descriptors and matching techniques which in combination showed an increased recognition rate compared to the systems being deployed in isolation. To deal with the increased complexity, we proposed a two-step verification approach that significantly reduced computation time compared to previous systems. All our code is publicly available.

Potential future work includes clustering of feature descriptors into codebooks to obtain prior probabilities of each descriptor, computing mesh models and rendered views of each object model to generate training views from a larger set of camera poses, data augmentation for the CNN pipeline, and using a hierarchical segmentation approach to take into account over- and undersegmentation. Also we are interested

$k_{\text{SIFT/SHOT/ESF/CNN}}$	δ_γ	$m^t_{\text{f, min}}/m^t_{\text{f, max}}$	v^t	c^t	λ_{m}	λ_Φ	λ_{s}	σ_{3D}	σ_{L} / σ_{AB}	ρ	δ_t / δ_r	$\tau_{\text{c,min}}/\tau_{\text{c,max}}$	$z^t_{\text{SIFT/SHOT/ESF/CNN}}$
3/1/3/3	60°	0.2/0.4	0.15	5	1	10^6	10	0.1	1000/30	1.5 σ_{3D}	2cm / 10°	0.5 / 1.2	2.5/1.5/2.5/2.5

TABLE I: Parameters used in evaluation.



(a) SIFT combinations (b) SHOT combinations (c) ESF combinations (d) CNN combinations

Fig. 3: Feature Evaluation on the Willow Dataset.

	precision	recall	fscore	time [s]
Aldoma *et al.* [1]	0.973	0.856	0.911	
Xie *et al.* [2]	0.983	0.878	0.927	38.1
Aldoma *et al.* [4]	0.943	0.709	0.809	5.2
Tang *et al.* [6]	0.888	0.648	0.749	20.0
Ours (SIFT only)	0.966	0.656	0.740	2.5
Ours (SHOT only)	0.830	0.284	0.367	2.4
Ours (CNN only)	0.716	0.546	0.489	20.4
Ours (ESF only)	0.678	0.347	0.358	13.4
Ours (SIFT+SHOT)	0.956	0.703	0.770	2.7
Ours (SIFT+ESF)	0.819	0.749	0.738	14.1
Ours (SIFT+CNN)	0.799	0.830	0.761	17.3
Ours (SHOT+ESF)	0.670	0.460	0.475	13.9
Ours (SHOT+CNN)	0.727	0.675	0.617	13.9
Ours (ESF+CNN)	0.745	0.592	0.529	32.5
Ours (SIFT+CNN)	0.799	0.830	0.761	20.0
Ours (SIFT+SHOT+ESF)	0.804	0.774	0.743	14.8
Ours (SIFT+SHOT+CNN)	0.791	0.871	0.781	21.6
Ours (SIFT+ESF+CNN)	0.835	0.753	0.736	27.5
Ours (SHOT+ESF+CNN)	0.685	0.486	0.505	24.8
Ours (SIFT+SHOT+ESF+CNN)	0.755	0.861	0.750	34.3

TABLE II: Recognition results on the Willow Dataset.

in the performance of the proposed method for partial object models as proposed in [21].

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS and No. 610532, SQUIRREL.

REFERENCES

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *PAMI*, 2015.

[2] Z. Xie, A. Singh, J. Uang, K. S. Narayan, and P. Abbeel, "Multimodal blending for high-accuracy instance recognition," in *IROS*. IEEE, 2013.

[3] J. Glover and S. Popovic, "Bingham procrustean alignment for object detection in clutter," in *IROS*. IEEE, 2013, pp. 2158–2165.

[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 *ICRA*. IEEE, 2013, pp. 2104–2111.

[5] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3d object recognition in cluttered scenes with local surface features: A survey," *PAMI*, vol. 36, no. 11, pp. 2270–2287, 2014.

[6] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *ICRA*. IEEE, 2012.

[7] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *IJRR*, 2012.

[8] A. Aldoma, F. Tombari, R. Rusu, and M. Vincze, "Our-cvfh: Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation," in *Joint DAGM-OAGM Pattern Recognition Symposium*, 2012.

[9] J. Prankl, A. Aldoma, A. Svedja, and M. Vincze, "Rgb-d object modelling for object recognition and tracking," in *IROS*. IEEE, 2015.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *ICIP*, 2011, pp. 809–812.

[12] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *ROBIO*. IEEE, 2011, pp. 2987–2992.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[14] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *IJCV*, vol. 116, no. 1, pp. 66–89, 2016.

[15] 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.

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

[17] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28 – 42, 2006.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] M. Maischberger, "COIN-OR METSlib, a metaheuristics framework in modern c++," http://www.coin-or.org/metslib/docs/releases/0.5.2/metslib-tr.pdf, April 2011.

[21] T. Fäulhammer, R. Ambrus, C. Burbridge, M. Zillich, J. Folkesson, N. Hawes, P. Jensfelt, and M. Vincze, "Autonomous learning of object models on a mobile robot," *IEEE Robotics and Automation Letters*, vol. 2, no. 99, pp. 26–33, 2016.