2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Multimodal Blending for High-Accuracy Instance Recognition

Ziang Xie, Arjun Singh, Justin Uang, Karthik S. Narayan, Pieter Abbeel

***Abstract*— Despite the rich information provided by sensors such as the Microsoft Kinect in the robotic perception setting, the problem of detecting object instances remains unsolved, even in the tabletop setting, where segmentation is greatly simplified. Existing object detection systems often focus on textured objects, for which local feature descriptors can be used to reliably obtain correspondences between different views of the same object.**

We examine the benefits of dense feature extraction and multimodal features for improving the accuracy and robustness of an instance recognition system. By combining multiple modalities and blending their scores through an ensemble-based method in order to generate our final object hypotheses, we obtain significant improvements over previously published results on two RGB-D datasets. On the Challenge dataset, our method results in only one missed detection (achieving 100% precision and 99.77% recall). On the Willow dataset, we also make significant gains on the prior state of the art (achieving 98.28% precision and 87.78% recall), resulting in an increase in F-score from 0.8092 to 0.9273.

I. Introduction

Object recognition remains a challenging problem in computer vision. Researchers often divide the recognition task into (1) category-level recognition, where objects in a particular category are given the same label (e.g. "bowl" or "soda can"), and (2) instance recognition, where each specific object is given its own label (e.g. "Coke can" or "Diet Coke can"). Several differences distinguish the robotic setting from the classic computer vision setting of detecting objects in 2D images. First, since robots in fixed environments may only have to interact with on the order of hundreds of objects, instance recognition may be sufficient for a wide variety of tasks. Second, in order for robots to be able to manipulate the recognized objects, the object poses must also be determined. Finally, given the relatively low cost of RGB-D sensors, such as the Microsoft Kinect, in comparison to other components of robotic platforms, both color images and point clouds are becoming standard in the robotic perception setting [1]. Despite the additional information provided by the depth channel, no instance recognition systems are robust enough to detect all objects in the standard benchmark settings we consider here, let alone in highly unstructured household and office environments.

In this paper, we present a system which significantly improves on the state of the art for the above task. While we briefly describe our end-to-end pipeline, we emphasize that our primary contributions are as follows:

1. We present experiments demonstrating that dense feature extraction (with moderate downsampling) results in significantly higher recall than feature extraction at keypoint-based interest points. This holds at training time, when building feature models, and at test time, when extracting features from the test images.
2. We illustrate how a discriminative extension of feature weighted linear stacking [2] can be used to generate object hypotheses using a learned combination of scores derived from texture, color, and shape-based features, leading to another significant boost in performance.
3. Finally, we achieve state-of-the-art results on two instance recognition datasets. On the Challenge dataset, our method results in near perfect performance, (1.000 precision and 0.9977 recall). On the Willow dataset, we present a significant leap over the previous state of the art, with .9828 precision and 0.8778 recall, corresponding to an increase in F-score from 0.8092 to 0.9273.

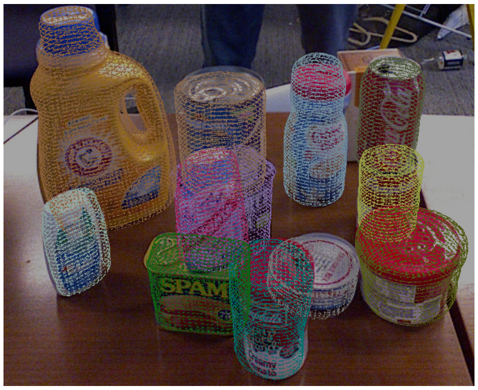


Fig. 1: Example detection. Despite clutter and heavy occlusions, our approach accurately captures object class and pose.

II. Related Work

Many existing approaches toward instance recognition first extract descriptors from a training set of objects, then match them to corresponding descriptors in a given test scene, using the correspondences to simultaneously determine the correct object and pose. One such example is the MOPED system of Collet et al. [3], which first constructs a sparse descriptor database by extracting SIFT features [4] at training time. At test time, MOPED uses the SIFT features to jointly estimate object class and pose. Unlike our system, it does not

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA. zxie@berkeley.edu, {arjun, karthik.narayan}@eecs.berkeley.edu, justin.uang@berkeley.edu, pabbeel@cs.berkeley.edu

978-1-4673-6358-7/13/$31.00 ©2013 IEEE

use depth information or perform multimodal verification. Aldoma et al. [5] obtain good performance by combining two pipelines, one using 2D and the other 3D features, via an optimization-based hypothesis verification framework. They too use features extracted at keypoints, and differ in their use of a 3D histogram-based feature.

While methods using the depth modality typically extract features from depth maps for use in recognition, such as extensions of histograms of oriented gradients (HOG) [6], [7], spin images [8], and point feature histograms [9], we use depth primarily for segmentation and to determine the 3D correspondences of pixels in test images.

Several works on category recognition suggest that dense sampling tends to yield higher performance [10], [11]. For example, Tuytelaars et al. [10] attribute part of the success of their Naive Bayes Nearest Neighbor classifier to dense descriptor computation. In the unsupervised feature learning context, Coates et al. [11] demonstrate that dense convolution in convolutional neural networks leads to higher performance. Nowak et al. also illustrate the importance of dense sampling for bag-of-features image classification [12]. To the best of our knowledge, however, this paper is the first to demonstrate that dense features play a crucial role in attaining strong performance on the instance recognition problem.

In addition to improving performance using dense features, we consider using an ensemble of models over several modalities to aid in recognition. Ensemble learning, also referred to as *blending*, has been explored thoroughly in the machine learning community [13], [14]. One simple method is to learn an affine combination of the scores obtained through each modality. Another method that extends this idea is feature weighted linear stacking [15], [16]. We evaluate both methods for the purpose of generating hypothesis detections.

Thorough experimentation illustrates that our approach obtains significantly better results than previous state of the art. Before presenting our approach, we first give an overview of the test datasets, which we use as the primary evaluation metrics.

III. Problem Setting

We evaluate our approach on two datasets, Challenge and Willow, which were collected for the 2011 ICRA Solutions in Perception Challenge. Both datasets share a set of 35 training objects, which must be detected in a variety of tabletop scenes.

The training data consists of 37 views of each training object, with each view containing a registered point cloud C, a 1024×1280 image I, the training object's image mask, and the associated camera matrix K for that view. At a high level, we build models of each object by merging information from each view. We present details of this training procedure in Section V.

The test data consists of a series of test scenes. Each test scene provides at most 20 scene views, each accompanied by a color image I and an associated point cloud C from a Kinect sensor. We also assume a gravity vector, which we compute using a single scene from both the Challenge and Willow test sets, since the sensor remains fixed throughout. Test scenes can be highly cluttered, potentially containing any subset of the 35 objects in any pose, as well as imposter objects not among the training objects. At test time, given a view with C, I, we are asked to jointly recognize each training object and its pose in each view independently, while ignoring imposter objects. The datasets differ greatly in difficulty; the Challenge dataset consists of 39 test scenes which may each contain multiple objects with low occlusions and no imposter objects, while the Willow dataset consists of 24 test scenes with heavy (possibly even full) occlusions as well as imposter objects. We present sample training and testing data from each dataset in Section 5.

Our current solution operates under the following assumptions, satisfied by both the Willow and Challenge datasets:

1. Training objects have relatively non-specular and non-transparent surfaces; RGB-D sensors using structured-light approaches (e.g. the Kinect) can, at best, generate noisy, incomplete point clouds of highly-specular or transparent objects.
2. Training objects contain texture: we use gradient-based descriptors to estimate training object poses.
3. Objects in test scenes are supported by a tabletop plane: this allows for simple test scene segmentation (Section V).

From here, we organize the paper as follows: we first present a brief overview of our system, which builds on Tang et al. [17]. We then present our main contributions, namely an examination of dense feature extraction, multimodal feature models and pose-based verification, and multimodal blending. Finally, we conclude with a series of experiments detailing the performance of our pipeline on the Challenge and Willow datasets.

IV. System Overview

At training time, we first merge point clouds from multiple views of each training object, then construct a mesh using Poisson reconstruction. After extracting features from images of each view of the object, we then project the features onto the mesh. The resulting descriptor list with associated 3D model coordinates constitutes a feature model. Section V-A details our feature computation approach, and Section V-B details the types of feature we use.

At test time, we first detect table planes in the scene using a RANSAC plane fitting approach, eliminating hypotheses using the gravity vector for the dataset and selecting the highest supporting plane. We then apply agglomerative clustering to points lying above this supporting plane. Next, we project each test cluster onto the images to form image masks and extract features from unmasked portions of the image. To obtain the 3D coordinates associated with each feature, we project features extracted from the masked image back onto the 3D point cloud.

Algorithm 1 describes the procedure we run on each test cluster to compute a RANSAC score and pose estimate for each candidate object. We then use Algorithm 2 to compute

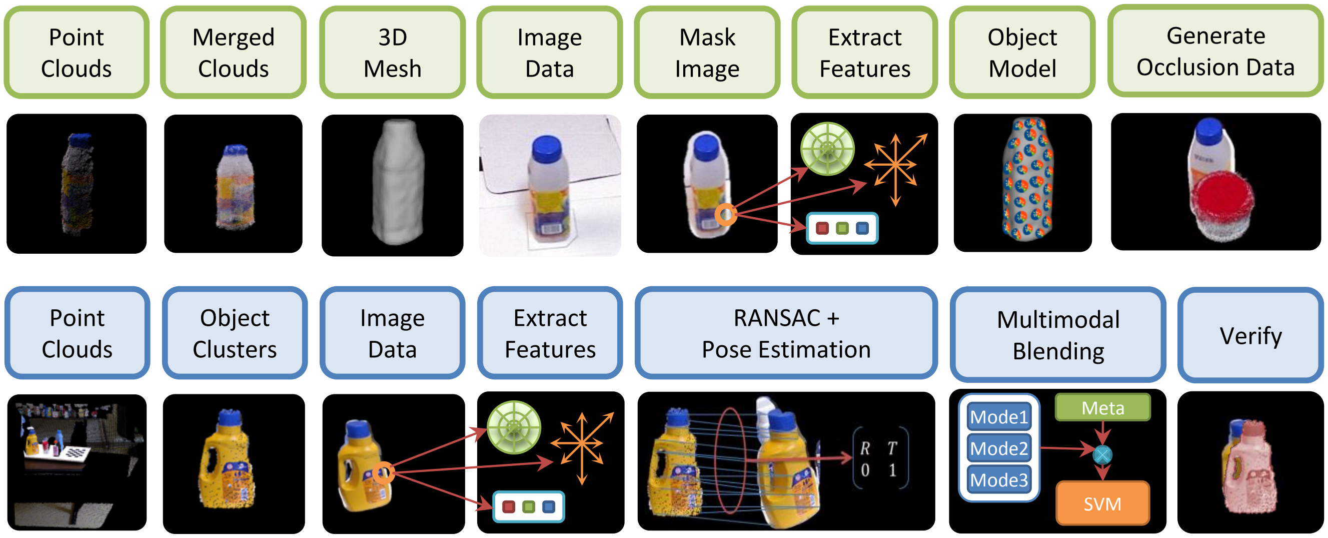


Fig. 2: An overview of our pipeline. The top row shows the training pipeline, and the bottom row shows the test pipeline.

Symbol		Description
I	∼	a color image
p	∼	a point (x, y, z)
P	∼	a point cloud of size $\|P\|$
f	∼	an image feature descriptor
F	∼	a set of descriptors of size $\|F\|$
M	∼	a feature model or (P, F) pair with each p_i corresponding to f_i
C	∼	a set of feature correspondences
$p(f)$	∼	point corresponding to descriptor f
$\mathrm{NN}(f, M^d)$	∼	nearest neighbor of feature f in model M^d, $d \in \{\text{shape, color, SIFT}\}$
ϵ	∼	distance threshold—either in feature space (ϵ_d) or 3D space ($\epsilon_{3\mathrm{D}}$)
s	∼	score—either RANSAC (s^{NN}) or verification (s^d)
$\hat{T}$	∼	an estimated 6DOF pose
N_{obj}	∼	number of training objects
N_{RANSAC}	∼	number of RANSAC iterations

TABLE I: Summary of Notation

pose-based verification scores (one for each type of feature) for each candidate object using their estimated poses.

Finally, we compute metafeatures on the test scenes (see Table II), and, given the metafeatures as well as the RANSAC and verification scores, we use a multimodal blender to output our final object hypotheses (Section V-C).

V. Methods

We now describe the key contributions of our approach, namely dense feature extraction, pose-based verification using multiple feature models, and multimodal blending.

Algorithm 1: RANSAC Pose Estimation

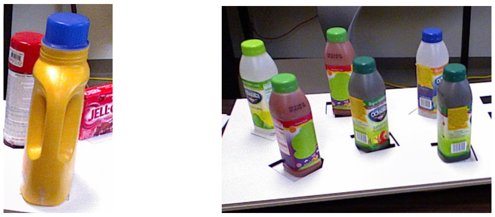
```
Input: test cluster $P^{\text{test}}$, corresponding test features
       $F^{\text{test}}$, feature models $\{M_i^d\}_{i=1}^{N_{\mathrm{obj}}}$ where $d$ is the
       model feature type, query-to-model NN
       correspondences $\{C_i\}_{i=1}^{N_{\mathrm{obj}}}$
Output: RANSAC scores $\{s_i^{\mathrm{NN}}\}_{i=1}^{N_{\mathrm{obj}}}$, estimated poses
        $\{\hat{T}_i\}_{i=1}^{N_{\mathrm{obj}}}$
Initialize $\{s_i^{\mathrm{NN}} = -\infty\}_{i=1}^{N_{\mathrm{obj}}}$
for $i = 1$ to $N_{\mathrm{obj}}$ do
    for $j = 1$ to $N_{\mathrm{RANSAC}}$ do
        $C_{ij} \leftarrow$ sample 3 correspondences from $C_i$
        $\hat{T}_{ij} \leftarrow$ EstimateTransform($C_{ij}$)
        Align $P^{\text{test}}$ to $M_i^d$ using $\hat{T}_i$
        for $j = 1$ to $|P^{\text{test}}|$ do
            if $\|p_j - p(\mathrm{NN}(f_j, M_i^d))\|_2 < \epsilon_{3\mathrm{D}}$ then
                $s_{ij}^{\mathrm{NN}} = s_{ij}^{\mathrm{NN}} + 1$        // Increment score
            end
        end
        if $s_{ij}^{\mathrm{NN}} > s_i^{\mathrm{NN}}$ then
            $s_i^{\mathrm{NN}} = s_{ij}^{\mathrm{NN}}$              // Update best score
            $\hat{T}_i = \hat{T}_{ij}$               // Update best pose
        end
    end
end
```

A. Dense Feature Extraction

We compute image features densely at both training and test time. At training time, rather than keep descriptors computed at every pixel, we employ voxel grid downsampling (with a leaf size of 0.5cm) of the descriptors for each view after projecting them onto the object mesh. At test time, we

Algorithm 2: Pose-based Verification

```
Input: test cluster $P^{\text{test}}$, corresponding test features
    $F^{\text{test}}$, estimated poses $\{\hat{T}_i\}_{i=1}^{N_{\text{obj}}}$, feature models
    $\{M_i^d\}_{i=1}^{N_{\text{obj}}}$, where $d$ is the feature model type
Output: verification scores $\{s_i^d\}_{i=1}^{N_{\text{obj}}}$
Initialize $\{s_i^d = 0\}_{i=1}^{N_{\text{obj}}}$
for $i = 1$ to $N_{\text{obj}}$ do
    Align $P^{\text{test}}$ to $M_i^d$ using $\hat{T}_i$
    for $j = 1$ to $|P^{\text{test}}|$ do
        // Radius search finds all model points
        // within $\epsilon_{3D}$ of $p_j^{\text{test}}$ in $M_i^d$ when aligned
        $F_{ij}^{\text{train}}, P_{ij}^{\text{train}} \leftarrow$ RadiusSearch($M_i^d$, $p_j^{\text{test}}$, $\epsilon_{3D}$)
        for $k = 1$ to $|P_{ij}^{\text{train}}|$ do
            if $\|f_j^{\text{test}} - f_{ijk}^{\text{train}}\|_2 < \epsilon_d$ then
                $s_i^d = s_i^d + 1$          // Increment score
                break
            end
        end
    end
end
```



(a) Example of nontextured test object view.

(b) Example test scene with only two non-imposters.

Fig. 3: Cases where our multimodal approach improves performance.

use a stride of 20px for the query image descriptors. Using these two methods hardly impacts performance due to high correlation between neighboring descriptors, and still yields approximately 5 to 10 times as many descriptors as using keypoints. See Section VI-D for a detailed analysis of the results using keypoints versus our dense sampling approach.

As described in Algorithm 1, we attempt to align each training object to a given test cluster using the SIFT feature model. Empirically, we find that the training object yielding the highest s^{NN} usually matches the testing object, assuming the test object is not an imposter.

In the presence of imposters and spurious segmentations, the ratio between the first and second highest s^{NN} is a reliable indicator of whether the first ranked object should be declared.[1] This baseline approach alone yields 100% precision and 99.31% recall on the Challenge dataset (which contains no imposters), but far from perfect performance on the Willow dataset (93.82% precision and 83.97% recall).

B. Multimodal Feature Models and Pose-Based Verification

Motivated by the remaining errors on the Willow dataset, we also construct models to exploit local color and shape information in addition to the models using gradient-based image descriptors. Concretely, we construct color feature models using L*a*b* values at each pixel, and shape feature models using shape context features computed in a similar fashion as described by Belongie et al. [18], using the additional depth information to scale the downsampled edge points before binning.

After detecting and estimating the pose of an object in a scene, we then perform a verification of the detection using each of the SIFT, shape, and color models (Algorithm 2). These scores ($\{s^{\text{color}}, s^{\text{SIFT}}, s^{\text{shape}}\}_{i=1}^{N_{\text{obj}}}$) provide additional information for whether to declare or eliminate a detection.

Intuitively, the multimodal approach leverages the strength of each feature type in different contexts. For example, few texture cues in an object view (e.g. Figure 3a) typically yield low RANSAC scores, rendering the ratio test unreliable. In this case, shape cues can provide information regarding correct object detection.

Color information also greatly helps in improving precision. Given that our procedure estimates an accurate pose, the color ratio check

$$\frac{s_i^{\text{color}}}{|F^{\text{test}}|} > r_{\text{color}}$$

then serves as a reliable indicator of whether object i is the correct object. One case in particular this check works well for is instances of different flavors, such as the Odwalla bottles shown in Figure 3b, where the algorithm cannot reliably distinguish objects using only gradient or local shape information.

Given object hypotheses for a test cluster, we apply the color ratio check described above as well as verify that the object hypothesis ranks at the top in at least half of the model scores before declaring a detection.

C. Multimodal Blending

There are many rules that could be used in addition to the color ratio threshold described in the previous section. Rather than relying on (typically tedious and labor-intensive) hand-engineering to generate such rules for combining the multiple modalities, we use a method inspired by the feature-weighted linear stacking (FWLS) approach proposed by Sill et al. [2]. This method blends the scores obtained using each model through a systematic, data-driven approach. Furthermore, this method can leverage the intuition that certain models may be more reliable in some settings than others.

The FWLS approach described by Sill et al. blends the model outputs by using *metafeatures*, which provide information about which models might be most reliable for a particular input scene. Rather than performing regression against the outputs of several models, the FWLS approach performs regression against all pairs of products of metafeatures and model outputs, as illustrated in Figure 4. For example, the median color saturation of the test image is one metafeature

[1] Specifically, we use $r_{\text{NN}} = 1.5$; if the ratio is lower than this than we do not declare a detection.

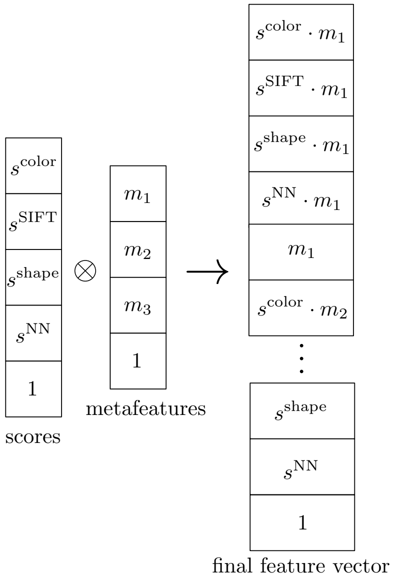


Fig. 4: Illustration of the computation of all pairs of products. The s^d values represent verification scores. The m_i values represent metafeatures. Note that for each test cluster, there will be N_{obj} such feature vectors—one per training object.

#	Description
1	median color saturation (HSV) of test image
2	fraction of pixels in test image where gradient magnitude > 10
3	visible ratio: (test cluster size)/(size of each M_{SIFT})
4	binary variable indicating whether object ranked first in RANSAC score
5	binary variable indicating whether object ranked first in color verification score
6	binary variable indicating whether object ranked first in shape verification score
7	binary variable indicating whether object ranked first in SIFT verification score

TABLE II: Metafeatures used for model score blending.

we use; low median color saturation suggests that color scores may be less reliable.

We extend the work of Sill et al. on FWLS in a regression setting to a discriminative setting in a method we refer to as *multimodal blending*. In short, we take pairwise products of metafeatures with model scores and use them as features for a classifier.

In particular, we use a standard ranking support vector machine (SVM) formulation to declare which object, if any, is present in each input cluster. Our formulation is given by

$$\begin{aligned} \underset{w,\ \xi}{\text{minimize}} \quad & \frac{\lambda}{2}\|w\|_2^2 + \sum_i \sum_{j \neq y_i} \xi_{ij} \\ \text{subject to} \quad & w^T \phi_{y_i}(x_i) \geq w^T \phi_j(x_i) + 1 - \xi_{ij}, \\ & \qquad \forall i, \forall j \neq y_i \end{aligned}$$

where w represents the weight parameters, the ξ_{ij} are slack variables, i ranges over all input clusters, j ranges over object labels, y_i is the correct label for input i, λ is a regularization parameter, x represents the values of all metafeatures and model scores for all objects for an input cluster, and ϕ_j denotes a feature function that constructs the feature vector (as illustrated in Figure 4) for object j.

During testing, the dot products between the weight vector and each of the feature vectors (i.e. $w^T \phi_j(x_i)$) yield a score for each object class, where the system declares the object with the highest score. In the presence of imposter objects, a threshold may be used such that the system can eliminate some spurious detections.

In order to provide training data to the SVM, we generate simulated occlusions on the Willow training data using our Challenge models as the occluding objects. We randomly sample occluding objects and poses around the ground truth pose provided for approximately ten views of each object in the Willow training data, resulting in approximately 10,000 simulated occlusions across all training objects. We then run our pipeline, treating the generated data as testing data, which gives the RANSAC and pose-based verification scores as well as the metafeatures for each view. We then use these scores and metafeatures as input training data for the SVM.

VI. Experiments and Analysis

A. Threshold Selection

As described in Section V, the pipeline contains two thresholds in the verification step: (1) the color ratio threshold and (2) the blending score threshold. Because the training data does not contain imposter objects, even though the Willow dataset does, we cannot reliably set these thresholds using a subset of the training data as a validation set. In previous work, this likely resulted in thresholds being tuned directly on the test set, leading to overfitting.

In order to avoid overfitting to the test set, we ensure that the thresholds for each scene are selected on data excluding that scene. Specifically, we use a leave-one-out procedure that chooses the best threshold from all scenes other than the scene currently being evaluated. Note that the Willow dataset consists of 24 scenes, with a varying number of objects and frames per scene. We run the system on all 23 scenes not under consideration, choose the threshold that results in the highest F-score on those 23 scenes, and then use it to evaluate the scene under consideration. Note that this procedure will almost always result in lower scores than what would be achieved by directly optimizing the F-score on all 24 scenes.

B. Single Instance Recognition

As a preliminary test, we evaluate our methods on recognizing individual objects. Following the setup used by Tang et al. [17], we use the Willow training data to build feature models and evaluate our system using the Challenge

Method	Precision	Recall	F-score
Tang et al. [17]	0.9672	0.9744	0.9710
Bo et al. [19]	0.9744	1.000	0.9870
Ours [no blending]	0.9976	0.9976	0.9980
Ours [blending]	**1.0000**	**1.0000**	**1.000**

TABLE III: Single object instance recognition. "No blending" declares the object with the highest RANSAC inlier count score, while "blending" uses the highest score (as output by the ranking SVM) with no metafeatures.

training data as test data. For this experiment, we remove any verification checks, simply choosing the highest scoring object as the detection. We present results in Table III, where we also compare our performance to the hierarchical matching pursuit algorithm described by Bo et al. [19]. Our method achieves perfect precision and recall on this task.

C. Multiple Instance Recognition

We now investigate our pipeline's performance on the Willow and Challenge testing data, which can contain multiple objects in the same frame. Recall that the Willow dataset may contain imposter objects not present in the training data (e.g. in Figure 3b); the system should not declare any detection on imposter objects.

We use both verification methods described in Section V-B for all results for this experiment. When we use blending, we also have a threshold for the blending score. We compare our results to those of Tang et al. [17] and Aldoma et al. [5] and surpass state-of-the-art performance on these datasets. We provide results for our method (1) without blending, (2) with blending but no metafeatures, and (3) with blending and metafeatures.

We present results for the Challenge dataset in Table IV. Note that Challenge is a small dataset, with only 434 objects to be detected. Without blending, our method already achieves near-perfect performance (perfect precision, 0.9931 recall), only failing to declare 3 out of the 434 correct detections. Although blending without metafeatures improves this further, adding metafeatures slightly descreases performance. We attribute this primarily to noise due to the dataset's small size, as only a small number of detections are changed.

We present results for Willow in Table V. On Willow, we present a significant leap over the previous state of the art, which we primarily ascribe to dense feature extraction and multimodal verification, yielding a recall of 0.8311 and a precision of .9976, corresponding to a significant increase in F-score (from 0.8092 to 0.9062). Even given this large performance increase, blending further increases performance by trading a small sacrifice in precision for a large improvement in recall. Incorporating all components (including blending and metafeatures) yields a recall of 0.8778 and precision of 0.9828, which correspond to a further increase in F-score to 0.9273. We analyze the remaining failure cases in Section VII.

Method	Precision	Recall	F-score
Tang et al. [17]	0.9873	0.9023	0.9429
Aldoma et al. [5]	0.9977	0.9977	0.9977
Ours [no blending]	1.0000	0.9931	0.9965
Ours [blending]	1.0000	0.9977	**0.9988**
Ours [blending+mf]	0.9954	0.9885	0.9919

TABLE IV: Results on Challenge dataset.

Method	Precision	Recall	F-score
Tang et al. [17]	0.8875	0.6479	0.7490
Aldoma et al. [5]	0.9430	0.7086	0.8092
Ours [no blending]	0.9976	0.8311	0.9062
Ours [blending]	0.9683	0.8827	0.9235
Ours [blending+mf]	0.9828	0.8778	**0.9273**

TABLE V: Results on Willow dataset.

D. Comparison to Using Sparse Keypoints

Our experiments indicate that dense feature extraction plays a major role in attaining high performance. We examine the effects of using SIFT keypoints versus our current approach. At training time, we extract features at each pixel, then perform voxel grid downsampling of the features after projecting them onto our mesh models. At test time, we downsample by using a stride of 20 over the pixels at which we extract descriptors.

In general, using keypoints results in good precision but significantly reduced recall. Table VI illustrates the effects on performance of using SIFT keypoints when training feature models and when extracting test image descriptors.

E. Blending with Keypoints

Although multimodal blending boosts performance even when applied on top of dense feature extraction, we observe that it yields significantly better relative increases in performance when using sparse keypoints at test time with query images.

Table VII shows the large increases in performance when applying blending to results obtained from RANSAC and multimodal verification with keypoints.

These results are largely due to the fact that many of the remaining errors when using dense feature extraction stem from poor pose estimations from the RANSAC phase, in which case the RANSAC and verification scores are unreliable (we discuss such failure cases further in Section VII). In contrast, when using keypoints, there are still many cases where a good pose was estimated, but not enough features were extracted to determine the object class using the ratio test with SIFT scores alone. In these cases, blending can combine scores across modalities, which significantly improves keypoint-based performance.

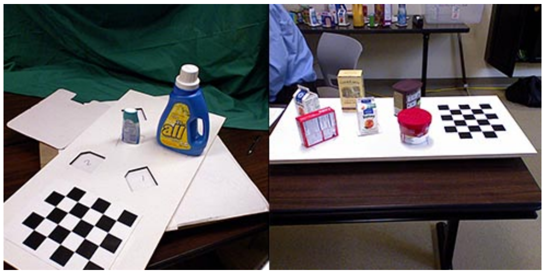


Fig. 5: Example test scenes from the Challenge and Willow datasets.

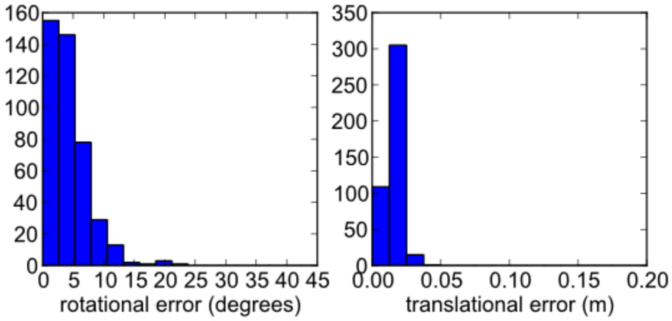


Fig. 6: Histograms of pose errors on Challenge dataset. Ground truth poses are unavailable for the Willow dataset.

Model	Query	Dataset	Prec.	Recall	F-score
sparse	sparse	Challenge	0.9894	0.8614	0.9210
sparse	sparse	Willow	0.9453	0.5412	0.6883
sparse	dense	Challenge	0.9879	0.9401	0.9634
sparse	dense	Willow	0.9279	0.7199	0.8108
dense	sparse	Challenge	0.9975	0.9171	0.9556
dense	sparse	Willow	0.9432	0.5915	0.7271
dense	dense	Challenge	1.0000	0.9931	**0.9965**
dense	dense	Willow	0.9382	0.8397	**0.8862**

TABLE VI: Performance using sparse vs. densely computed (then downsampled) SIFT models and query features. Only RANSAC scores and the ratio test are used for these results.

Experiment	Prec.	Recall	F-score
Challenge	0.9975	0.9171	0.9556
Challenge [blending]	0.9881	0.9585	0.9731
Challenge [blending+mf]	0.9905	0.9585	0.9742
Willow	0.9432	0.5915	0.7271
Willow [blending]	0.9508	0.7604	0.8451
Willow [blending+mf]	0.9475	0.7654	0.8468

TABLE VII: Results when using keypoints at test time with blending.

F. Timing Results

All timing experiments were done on a commodity desktop with a 4-core i7 processor and 32GB of RAM.

The entire training phase takes under 6 minutes for a single object. By parallelizing across objects, we can complete the training phase for all 35 objects in well under an hour. Training the weight vector for multimodal blending on the 10,000 generated examples takes roughly 75s.

A timing breakdown for the testing phase (averaged over scenes in the Challenge test set) is given in Table VIII. Applying blending using a linear SVM simply consists of a dot product and thus takes a negligible amount of time. Note that all steps following segmentation (i.e. RANSAC and pose-verification scores) can be run in parallel across clusters.

It is possible to sacrifice some performance to speed up the testing phase by excluding the pose-based verification step, which yields the already state-of-the-art results given in Table VI. Another alternative to greatly speed up the testing phase is to combine keypoints with blending, which, as described in Section VI-E, yields good performance as well (the RANSAC and verification phases take $<$ 2s total when using keypoints).

G. Improving the Willow Dataset

A small number of failures on Willow are due to errors in the ground truth, where labeled objects are actually fully occluded. After fixing all such labeling errors, we report performance on this dataset as “Willow-Vis.”

Additionally, we created another set of ground truth labels for Willow, where only objects determined to be at least 20% visible are marked as present in the scene. We report performance on this dataset as “Willow-20.”

Both of these modified datasets are available for download at `http://rll.berkeley.edu/2013_IROS_ODP`. To give an idea of how many errors are due to the object being either fully or highly occluded, we report results for both datasets in Table IX.

VII. Discussion

We now discuss the two primary failure cases of our system on the Willow dataset, namely detection errors due to poor pose estimation and imposter objects being mistaken for training objects.

A. Pose Estimation Failures

We attribute the majority of the remaining missed detections to RANSAC failing to discover the correct pose for the correct object. When RANSAC fails, the verification scores are usually unreliable, leading to the algorithm declaring the incorrect object (or no object). Because RANSAC only works with local SIFT features, this frequently happens with highly occluded objects or when only a nontextured part of the object is visible. Incorporating features that are computed over larger windows or that are more robust for untextured objects into the RANSAC computation may eliminate many of these errors, although at present it is unclear how to best incorporate these into our framework.

Testing Step	Time Per Scene (s)
Segmentation	5.4
Feature extraction, SIFT/shape/color	5.1 / 5.4 / 0.3
RANSAC pose estimation	13.9
Verification, SIFT/shape/color	3.8 / 0.4 / 3.7
Total	38.1

TABLE VIII: Timing results, test phase.

Method	Precision	Recall	F-score
Willow [no blending]	0.9976	0.8311	0.9062
Willow [blending]	0.9683	0.8827	0.9235
Willow [blending+mf]	0.9828	0.8778	**0.9273**
Willow-Vis [no blending]	0.9898	0.8504	0.9148
Willow-Vis [blending]	0.9683	0.9032	0.9346
Willow-Vis [blending+mf]	0.9882	0.8982	**0.9386**
Willow-20 [no blending]	0.9963	0.8780	0.9334
Willow-20 [blending]	0.9677	0.9319	0.9494
Willow-20 [blending+mf]	0.9690	0.9325	**0.9504**

TABLE IX: Results on modified Willow datasets. "Willow-Vis" refers to a ground truth labeling in which objects that are fully occluded are not counted as present in the scene. "Willow-20" refers to a ground truth labeling in which only objects that are hand-labeled as at least 20% visible (by human inspection) are counted as present in the scene. Results on Willow are repeated for ease of comparison.

B. Failures Due to Imposters

There are also a small number of errors due to imposter objects being declared as one of the training objects. Because the training data contains no imposter objects, the classifier cannot differentiate between the training objects and an imposter that has a high score for a single feature model, but only moderate scores for the other feature models. Adding imposter objects to the training data, which could be used as negatives for the classifier, may help eliminate these failure cases. Imposters would not require complete models; a collection of views without pose information would suffice.

VIII. Conclusion

This paper presents several methods which together yield significantly higher precision and recall on the Challenge and Willow datasets compared to the prior state of the art. The key contributing factors to our approach's performance are dense feature extraction, multimodal feature models, and data-driven blending of the scores for each feature model according to metafeatures of the current scene.

For visualizations of every detection (and mistake) made by our algorithm, please refer to:

http://rll.berkeley.edu/2013_IROS_ODP

To allow for comparison on a version of the dataset with completely occluded objects removed from the labeling (and another with mostly occluded objects removed), we provide supplementary ground truth labelings for Willow at the above website. A significant number of errors are due to objects being completely occluded in the supplied views.

Acknowledgements

We thank Jie Tang and James Sha for insightful discussions. This work is supported in part by ONR Grant # N00014-12-1-0756 and by ONR YIP Award # N00014-13-1-0570. Arjun Singh is supported by an NDSEG Fellowship. Karthik Narayan is supported by an NSF Graduate Fellowship.

References

[1] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on on Robotics and Automation (ICRA)*, 2011.

[2] Joseph Sill, Gábor Takács, Lester Mackey, and David Lin. Feature-weighted linear stacking. *CoRR*, abs/0911.0460, 2009.

[3] M. Martinez, A. Collet, and S. S. Srinivasa. MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System. In *International Conference on on Robotics and Automation (ICRA)*, 2010.

[4] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision (IJCV)*, 2004.

[5] 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 *International Conference on Robotics and Automation*, 2013.

[6] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *International Conference on on Robotics and Automation (ICRA)*, 2011.

[8] A. Johnson and M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. In *Pattern Analysis and Machine Intelligence (TPAMI)*, 1999.

[9] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *International Conference on on Robotics and Automation (ICRA)*, 2009.

[10] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN Kernel. In *International Conference on Computer Vision (ICCV)*, 2011.

[11] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[12] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *European Conference on Computer Vision (ECCV)*, 2006.

[13] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK, UK, 1995. Springer-Verlag.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[15] Z. Marton, F. Seidel, F. Balint-Benczedi, and M. Beetz. Ensembles of Strong Learners for Multi-cue Classification. *Pattern Recognition Letters (PRL), Special Issue on Scene Understandings and Behaviours Analysis*, 2012. In press.

[16] J. Sill, G. Takács, L. Mackey, and D. Lin. Feature-weighted linear stacking. *CoRR*, abs/0911.0460, 2009.

[17] J. Tang, S. Miller, A. Singh, and P. Abbeel. A Textured Object Recognition Pipeline for Color and Depth Image Data. In *International Conference on on Robotics and Automation (ICRA)*, 2012.

[18] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. In *Pattern Analysis and Machine Intelligence (TPAMI)*, 2002.

[19] L. Bo, X. Ren, and D. Fox. Unsupervised Feature Learning for RGB-D Based Object Recognition. In *International Conference on Experimental Robotics (ISER)*, June 2012.