

Towards Real-time Object Recognition and Pose Estimation in Point Clouds

Marlon Marcon¹^a, Olga Regina Pereira Bellon²^b and Luciano Silva²^c

¹*Department of Software Engineering, Federal University of Technology - Paraná, Dois Vizinhos, Brazil*

²*Department of Computer Science, Federal University of Paraná, Curitiba, Brazil*

Keywords: Transfer Learning, 3D Computer Vision, Feature-based Registration, ICP Dense Registration, RGB-D Images.

Abstract: Object recognition and 6DoF pose estimation are quite challenging tasks in computer vision applications. Despite efficiency in such tasks, standard methods deliver far from real-time processing rates. This paper presents a novel pipeline to estimate a fine 6DoF pose of objects, applied to realistic scenarios in real-time. We split our proposal into three main parts. Firstly, a Color feature classification leverages the use of pre-trained CNN color features trained on the ImageNet for object detection. A Feature-based registration module conducts a coarse pose estimation, and finally, a Fine-adjustment step performs an ICP-based dense registration. Our proposal achieves, in the best case, an accuracy performance of almost 83% on the RGB-D Scenes dataset. Regarding processing time, the object detection task is done at a frame processing rate up to 90 FPS, and the pose estimation at almost 14 FPS in a full execution strategy. We discuss that due to the proposal's modularity, we could let the full execution occurs only when necessary and perform a scheduled execution that unlocks real-time processing, even for multitask situations.


1 INTRODUCTION


Object recognition and 6D pose estimation represent a central role in a broad spectrum of computer vision applications, such as object grasping and manipulation, bin picking tasks, and industrial assemblies verification (Vock et al., 2019). Successful object recognition, highly reliable pose estimation, and near real-time operation are essential capabilities and current challenges for robot perception systems.


A methodology usually employed to estimate rigid transformations between scenes and objects is centered on a feature-based template matching approach. Assuming we have a known item or a part of an object, this technique involves searching all the occurrences in a larger and usually cluttered scene (Vock et al., 2019). However, due to natural occlusions, such occurrences may be represented only by a partial view of an object. The template is often another point cloud, and the main challenge of the template matching approach is to maintain the runtime feasibility and preserve the robustness.

Template matching approaches rely on RANSAC-based feature matching algorithms, following the pipeline proposed by (Aldoma et al., 2012b). RANSAC algorithm has proven to be one of the most versatile and robust. Unfortunately, for large or dense point clouds, its runtime becomes a significant limitation in several of the example applications mentioned above (Vock et al., 2019). When we seek a 6Dof estimation pose, the real-time is a more challenging task (Marcon et al., 2019). In an extensive benchmark of full cloud object detection and pose estimation, (Hodan et al., 2018) reported runtime of a second per test target on average.

Deep learning strategies for object recognition and classification problems have been extensively studied for RGB images. As the demand for good quality labeled data increases, large datasets are becoming available, serving as a significant benchmark of methods (deep or not) and as training data for real applications. ImageNet (Deng et al., 2009) is, undoubtedly, the most studied dataset and the *de-facto* standard on such recognition tasks. This dataset presents more than 20,000 categories, but a subset with 1,000 categories, known as ImageNet Large Scale Visual Recognition Challenge (ILSVRC), is mostly used.

^a <https://orcid.org/0000-0001-9748-2824>

^b <https://orcid.org/0000-0003-2683-9704>

^c <https://orcid.org/0000-0001-6341-1323>

Training a model on ImageNet is quite a challenging task in terms of computational resources and time consumption. Fortunately, transferring its models offer efficient solutions in different contexts, acting as a blackbox feature extractor. Studies like (Agrawal et al., 2014) explore and corroborate this high capacity of transferring such models to different contexts and applications. Regarding the use of pre-trained CNN features, some approaches handle the object recognition on the Washington RGB-D Object dataset, e.g., (Zia et al., 2017) with the VGG architecture and (Caglayan et al., 2020) evaluate several popular deep networks, such as AlexNet, VGG, ResNet, and DenseNet.

This paper introduces a novel pipeline to deal with point cloud pose estimation in uncontrolled environments and cluttered scenes. Our proposed pipeline recognizes the object using color feature descriptors, crops the selected bounding-box reducing the scenes' searching surface, and finally estimates the object's pose in a traditional local feature-based approach. Despite adopting well-known techniques in the 2D/3D computer vision field, our proposal's novelty centers on the smooth integration between 2D and 3D methods to provide a solution efficient in terms of accuracy and time.

2 BACKGROUND

Recognition systems work with objects, which are digital representations of tangible real-world items that exist physically in a scene. Such systems are unavoidably machine-learning-based approaches that use features to locate and identify objects in a scene reliably. Together with the recognition, another task is to estimate the location and orientation of the detected items. In a 3D world, we estimate six degrees of freedom (6DoF), which refers to the geometrical transformation representing a rigid body's movement in a 3D space, i.e., the combination of translation and rotation.

2.1 Color Feature Extraction

As a mark on the deep learning history, (Krizhevsky et al., 2012) presented the first Deep Convolutional Architecture employed on the ILSVRC, an 8-layer architecture dubbed AlexNet. This network was the first to prove that deep learning could beat hand-crafted methods when trained on a large scale dataset. After that, ConvNets became more accurate, deeper, and bigger in terms of parameters. (Simonyan and Zisserman, 2015) propose VGG, a network that dou-

bled the depth of AlexNet, but exploring tiny filters (3×3), and became the runner-up on the ILSVRC, one step back the GoogLeNet (Szegedy et al., 2015), with 22 layers. GoogLeNet relies on the Inception architecture (Szegedy et al., 2016). Another type of ConvNets, called ResNets (He et al., 2016), uses the concept of residual blocks that use skip-connection blocks that learn residual functions regarding the input. Many architectures have been proposed based on these findings, such as ResNet with 50, 101, and 152 (He et al., 2016). Also, based on developments regarding the residual blocks, (Xie et al., 2017) developed the ResNeXt architecture. The basis upon ResNeXt blocks resides on parallel ResNet-like blocks, which have the output summed before the residual calculation. Some architectures propose the use of Deep Learning features on resource-limited devices, such as smartphones and embedded systems. The most prominent architecture is the MobileNet (Sandler et al., 2018). Another family of leading networks is the EfficientNet (Tan and Le, 2019). Relying on the use of these lighter architectures, EfficientNet proposes very deep architectures without compromise resource efficiency.

2.2 Pose Estimation

As presented in (Aldoma et al., 2012b), a comprehensive registration process usually consists of two steps: coarse and fine registrations. We can produce a coarse registration transformation by performing a manual alignment, motion tracking or, the most common, by using the local feature matching. Local-feature-matching-based algorithms automatically obtain corresponding points from two or multiple point-clouds, coarsely registering by minimizing the distance between them. These methods have been extensively studied and have confirmed to be compliant and computer efficient (Guo et al., 2016). After coarsely registering the point clouds, a fine-registration algorithm is applied to refine the initial coarse registration iteratively. Examples of fine-registration algorithms include the ICP algorithm that perform point-to-point alignment (Besl and McKay, 1992), or point-to-plane (Chen and Medioni, 1992). These algorithms are suitable for matching between point-clouds of isolated scenes (3D registration) or between a scene and a model (3D object recognition). This proposal adopted two approaches to generate the initial alignment: a traditional feature-based RANSAC and the Fast Global Registration (FGR) (Zhou et al., 2016).

3 PROPOSED APPROACH

In this section, we explain in detail our proposed approach. Our proposed pipeline starts from an RGB image and its corresponding point cloud, generated from RGB and depth images. These inputs are submitted to our three-stage architecture: color feature classification, feature-based registration, and fine adjustment. We depict our proposal in Figure 1 and present these steps in the next sections.

3.1 Color Feature Classification

Our proposal starts detecting the target object and estimating a bounding box of it. After this detection, we preprocess the image and submit to a deep-learning-based color feature extractor. The preprocessing step includes image cropping and resizing to adjust to the network input dimensions. The deep network architectures employed in our experiments output a feature vector, 1000 bins long, used to predict the object's instance, by a pre-trained ML classifier. We emphasize that our approach is size-independent regarding the feature vector, but for a fair comparison we chose networks with the same output size.

In our trials, we explored the achievements of Table 2, and selected the most accurate networks: ResNet101 (He et al., 2016), MobileNet v2 (Sandler et al., 2018), ResNeXt101 32×8d (Xie et al., 2017), and EfficientNet-B7 (Tan and Le, 2019). These networks input a 224×224 pixel image and output a 1000 bins feature vector. We employed the Logistic regression classifier, chosen after a performance evaluation of standard classifiers, to name: Support Vector Classifier (SVC) with linear and radial-based kernels, Random forest, Multilayer perceptron, and Gaussian naïve Bayes. We explore two variants of our ML model: a pre-trained on the Washington RGB-D Object dataset, and a distinct model, also in such dataset, but with a reduced number of objects, i.e., those annotated on the Washington RGB-D Scenes dataset. The latter provides an application-oriented approach, reducing the number of achievable classes, the inference time, and model size (Table 4). To verify the best accurate classifier, we do not perform object detection. Instead, we get the ground-truth bounding boxes provided by the dataset, hence verifying for each ML system which is the best feasible performance.

3.2 Feature-based Registration

We build a model database by extracting and storing useful information about the objects in a previous

step. The database is composed of information concerning each item, as well as the extracted features of them. We choose a local-descriptors-based approach to estimate the object's pose. For each instance of an object, we store several partial views of it. Between these views, our method will select the most likely to the correspondent object on the scene.

Based on the predicted objects' classes, we can select a set of described views from the models' database. We then perform a feature-based registration between these views and the point cloud of the scene's object (previously cropped based on the detected bounding box). This method will estimate a transformation based on the correspondences between a scene and a partial view of an object. Then, the view with the highest number of inliers and at least three correspondences is selected. The estimated affine transformation will be input to the ICP algorithm and perform a dense registration.

We process each cloud with a uniform sampling as a keypoint extractor, adopting a leaf size of 1 cm. After, we describe each keypoint using the FPFH (Rusu et al., 2009) descriptor with a radius of 5 cm. We choose this descriptor due to its processing time and size (33 bins), well-suited for real-time applications. Methods like CSHOT (Salti et al., 2014) describes the color and geometric information and has proven to be an accurate solution applied to single object recognition on RGB-D Object dataset (Ouardi et al., 2016). However, with a descriptor length of 1344 bins, it is not suitable for real-time feature-matching. Another proposal that deals with color is PFHRGB (Rusu et al., 2008), which, despite being shorter (250 bins) than CSHOT, presents inefficient calculation time (Marcon et al., 2019).

To perform the coarse registration step, we test two methods previously presented: RANSAC and FGR. We considered for both techniques an inlier correspondence distance lower than 1 cm between scene and models. We set the convergence criteria for RANSAC to 4M iterations and 500 validation steps, and for FGR to 100 iterations, following (Choi et al., 2015) and (Zhou et al., 2016).

3.3 Fine-adjustment

The previous step outputs an affine transformation that could work as a final pose of the object concerning the scene. However, to guarantee a fine-adjustment, we employ an additional step to the process. We adopt the ICP algorithm, based on the point-to-plane approach (Chen and Medioni, 1992), to perform a dense registration. We use the transformation resultant from the registration step, the scene, and

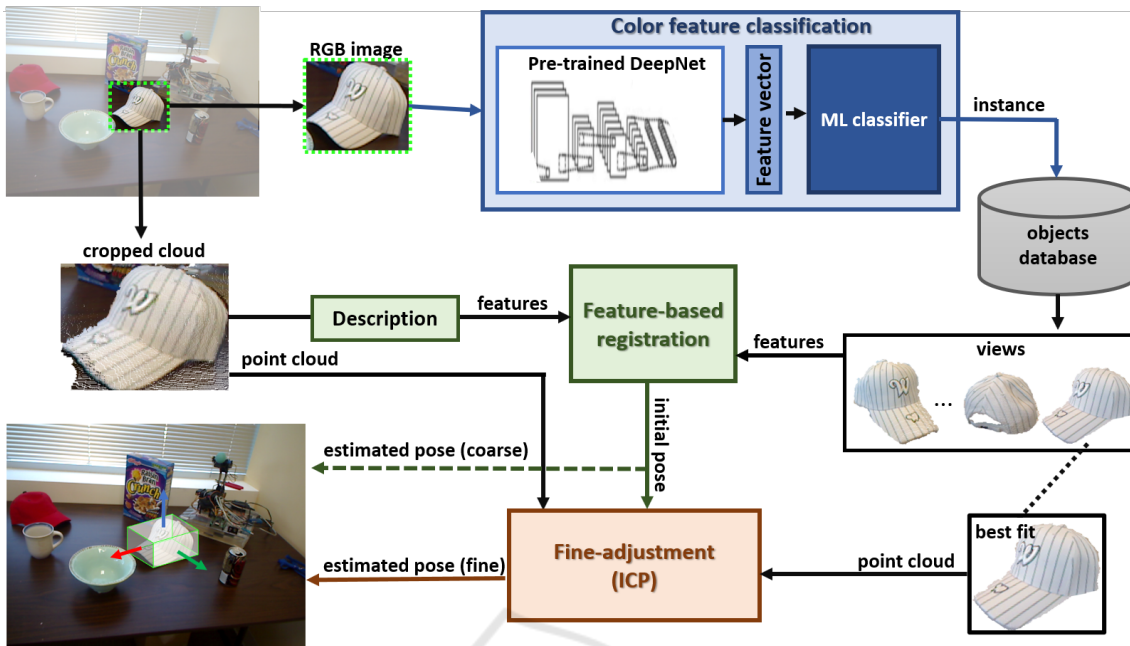


Figure 1: Pipeline of the proposed approach to pose estimation. To estimate the pre-segmented object’s instance, we extract its features by a deep learning color-based extractor and a pre-trained ML classifier. After selecting the objects database, the view with the highest number of correspondences resulting from a feature-based registration algorithm. Finally, we apply an ICP dense registration algorithm to estimate the position and pose of the object.

best-fitted view clouds as input. We set the maximum correspondence distance threshold to 1 cm. It is important to point that again, our proposal is generic, and the fine adjustment algorithm employed in this stage is flexible. Methods such as ICP point-to-point (Besl and McKay, 1992) and ColoredICP (Park et al., 2017) are perfectly adapted to our pipeline.

4 EXPERIMENTAL RESULTS

4.1 Dataset

We validate our proposal on the Washington RGB-D Object and Scenes datasets. Proposed by (Lai et al., 2011a) the RGB-D Object contains a collection of 300 instances of household objects, grouped in 51 distinct categories. Each object includes a set of views, captured from different viewpoints with a Kinect sensor. A collection of 3 images, including RGB, depth, and mask is presented for each view. In total, this dataset has about 250 thousand distinct images. The authors also provide a dataset of scenes, named RGB-D Scenes. This evaluation dataset has eight video sequences of every-day environments. A Kinect sensor positioned at a human eye-level height acquires all the images at a 640×480 resolution. This dataset is related to the first one, composed of 13 of the 51 object

categories on the Object dataset. These objects are positioned over tables, desks, and kitchen surfaces, cluttered with viewpoints and occlusion variation, and have annotation at category and instance levels. A bidimensional bounding box represents the ground-truth of each object’s position. Figure 2 presents examples of both datasets. Table 1 gives some details regarding the name and size of the sequences, and their average number of objects.



Figure 2: Examples of models and scenes from the Washington RGB-D Scenes dataset (top row), and objects from the RGB-D Object dataset (bottom row). Source: Adapted from (Lai et al., 2011a).

4.2 Evaluation Protocol

We evaluate our proposal, quantitatively, and qualitatively. First, we consider CNN feature extraction and classification accuracy based on the models trained in the Object dataset (Table 2). We also verify the entire

Table 1: Details regarding the RGB-D Scenes datasets.

Scene	Number of frames	Models per frame
desk_1	98	1.89
desk_2	190	1.85
desk_3	228	2.56
kitchen_small_1	180	3.55
meeting_small_1	180	8.79
table_1	125	5.92
table_small_1	199	3.68
table_small_2	234	2.89
Average	179.25	3.89

dataset’s processing time, looking at the frame processing rate in both classification and pose estimation scenarios.

As the Scenes dataset does not provide ground-truth annotations concerning the objects’ pose, we had to find a plausible metric to evaluate the registration results. We adopted two different metrics: the Root mean squared error (RMSE) and an inlier ratio measurement. The latter represents the overlapping area between the source (model) and the target (scene). It is calculated based on the ratio between inlier correspondences and the number of points on the target. We also evaluate the correctness of predictions, both of object presence and pose. To do so, we follow (Marcon et al., 2019) and employ the Intersection over Union metric (IoU), defined as:

$$IoU = \frac{BB_{GT} \cap BB_{Est}}{BB_{GT} \cup BB_{Est}} \quad (1)$$

we consider BB_{GT} the 3D projection of the 2D bounding box, provided as ground-truth. BB_{Est} refers to the 3D bounding box that circumscribes the selected object view after applying the resulting transformation.

We found experimentally that, for this particular dataset, when we estimate the IoU between the object 3D BB and the scene 2D projection, often the former is fully contained in the latter. However, due to their sizes, the calculated IoU is too low. Hence, we consider another metric, which we call Model Intersection Ratio (MIR) that represent the intersection volume over the model estimation volume:

$$MIR = \frac{BB_{GT} \cap BB_{Est}}{BB_{Est}} \quad (2)$$

With the MIR metric, we guarantee that despite the IoU, when the estimation transform places the object inside (or nearly inside) the ground-truth 3D projection, a successful detection is performed. We consider a true positive when the $IoU \geq 0.25$ or the $MIR \geq 0.90$.

We compared our proposal with the standard 3D object recognition and pose estimation pipeline (Aldoma et al., 2012b), and with a boosted version of such pipeline, proposed by (Marcon et al., 2019). To calculate precision-recall curves (PRC), we varied the threshold on the minimum geometrically consistent correspondences, starting from at least three, related to each object’s best-suited partial view. The area under the PRC curve (AUC) is then calculated and provides comparative results that assess our proposals’ efficiency against traditional approaches.

4.3 Implementation Details

We performed our tests on a Linux Ubuntu 18.04 LTS machine, equipped with a CPU Ryzen 7 2700X, 32GB of RAM, and a GPU Geforce RTX 2070 Super. To process the point clouds, perform keypoint extraction, description with FPFH, and registration with RANSAC and FGR, we used the Open3D Library. We preprocess images using Pillow and OpenCV. Deep learning models were implemented in PyTorch, and the pre-trained models extracted from torchvision. To implement traditional and boosted versions of object recognition and pose estimation pipelines, we use PCL 1.8.1, OpenCV 3.4.2, and the saliency detection of (Hou et al., 2017), following (Marcon et al., 2019).

4.4 Results

This section summarizes the Washington RGB-D Scenes’ experimental evaluation results in terms of accuracy and processing time.

4.4.1 Object Detection Benchmark

To assess the generalization capacity of CNN pre-trained models, we perform an object detection evaluation on the Object dataset (Lai et al., 2011a). Table 2 present results regarding classification of partial views of objects. We evaluate the instance recognition scenario, following (Lai et al., 2011a), i.e., considering Alternating contiguous frame (ACF) and Leave-sequence-out (LSO) scenarios. We compared our results with state-of-the-art object detection methods on this dataset. We perceived that pre-trained networks provide reliable results as off-the-shelf color feature extractors. In both evaluation approaches, tested networks present competitive results concerning the other competitors. In LSO, ResNet101 (He et al., 2016) features figures in the third position, and in ACF, 5 of 7 architectures outperform previous proposals.

Table 2: Comparison of CNN color features on the Washington RGB-D Object dataset. The best result reported in blue, the second best in green, and the third in red.

Method	LSO	ACF
Lai <i>et al.</i> (RF) (Lai et al., 2011a)	59.9	90.1 \pm 0.8
Lai <i>et al.</i> (kSVC) (Lai et al., 2011a)	60.7	91.0 \pm 0.5
IDL (Lai et al., 2011b)	-	54.8 \pm 0.6
SP+HMP (Bo et al., 2013)	92.1	-
Multi-Modal (Schwarz et al., 2015)	92.0	-
MDSI-CNN (Asif et al., 2017)	97.7	-
MM-LRF-ELM (Liu et al., 2018)	91.0	-
HP-CNN (Zaki et al., 2019)	95.5	-
AlexNet (Krizhevsky et al., 2012)	89.8	93.9 \pm 0.4
ResNet101 (He et al., 2016)	94.1	95.3 \pm 0.3
VGG16 (Simonyan and Zisserman, 2015)	88.8	91.0 \pm 0.6
Inception v3 (Szegedy et al., 2016)	88.1	90.3 \pm 0.4
MobileNet v2 (Sandler et al., 2018)	93.8	95.8 \pm 0.3
ResNeXt101 32 \times 8d (Xie et al., 2017)	93.9	95.7 \pm 0.4
EfficientNet B7 (Tan and Le, 2019)	93.8	95.6 \pm 0.5

Despite the significant results, this evaluation is essential to select the most suitable to perform object recognition in realistic scenarios, such as those presented by the Scenes dataset (Lai et al., 2011a). As the trials’ output, we selected the top-four architectures to apply in our proposed pipeline.

4.4.2 Object Recognition in Real-world Scenes

We opposed the selected CNN architectures examining only a classification based on the RGB information, taking the annotated bounding box, and submitting to the *Color Feature Classification* stage of our pipeline (as in Section 3.1). Table 3 relates to instance-level recognition.

The first outcome of this evaluation is the dominance of two networks over the other competitors considering different aspects. EfficientNet (Tan and Le, 2019) architecture outperforms in terms of accuracy, and MobileNet v2 (Sandler et al., 2018) in terms of processing time w.r.t. the others in almost all scenes.

EfficientNet reaches an average accuracy of almost 67%, followed by MobileNet v2, with almost 53%. However, when we aim efficiency in processing time, EfficientNet does not perform so well, being the slowest network with a frame-rate of 3.02 per second. On the other hand, the MobileNet v2 fulfills the network’s main proposal to be time-efficient and accurate for embedded applications. It presents the second-best accuracy and the best frame-rate, with almost 7 FPS.

The full-set of the Object dataset contains 51 categories and 300 distinct instances. Concerning the Scenes dataset, the number of annotated samples drops to 6 categories and 22 instances, i.e., only a small set of objects of Object dataset is achievable on the Scenes dataset. When we use a model trained on the full-set, most categories or instances will never be detected. Thus, we learned a lighter classifier that considers only such specific instances (Table 4).

After this change on the model specificity, we distinguish a noticeable improvement in accuracy and the processing time, achieving MobileNet v2 a near real-time performance on average. A significant gain on accuracy was established, with over 10% for every architecture, pulling the best result to 83% for EfficientNet.

Regarding the frame processing rate, it is essential to notice that the average number of models varies from 1.85 to 8.79 over the scenes, with almost four objects per frame in mean (Table 3). Thus, we can infer that our proposal can deliver a near-real-time FPS, inclusive in a multi-classification problem. When we consider only a single target, the performance is almost four times faster, as presented in Table 7, on the *Color only* column.

4.4.3 Pose Estimation Results

Based on the assumption that we mapped the objects we aim to detect in a real-world scenario, we adopted those models trained on the RGB-D Object dataset subset. We considered only the instance detection situation. The reason for disregarding categories is that we could have intra-class misclassifications, corrupting the pose alignment step. For each instance detected by the *Color feature classification* stage, we take ten views of the referred object from the models’ database.

In Table 5 we report an evaluation concerning the Feature-based registration and Fine-adjustment stages of our pipeline. Getting a set of ten randomly selected views of the same object, we perform a coarse estimation by using RANSAC or FGR. We evaluate quantitatively such methods concerning the inlier ratio, RMSE, and execution time. We apply the resulting affine transformation as the input of an ICP dense registration and evaluate if this input can imply differences in the processing time.

Indeed, the FGR method is much faster than RANSAC. However, we observe that for both metrics RANSAC outperforms it. The Inlier ratio presented by the latter is around 50% higher than the faster method and also shows an RMSE more consistent. The transformation generated by the coarse alignment algorithm also impacts the ICP execution

Table 3: Instance classification performance on the RGB-D Scenes datasets.

Scene	MobileNet v2		Resnet101		ResNeXt101 32x8d		EfficientNet-B7	
	Acc	FPS	Acc	FPS	Acc	FPS	Acc	FPS
desk_1	42.70%	13.03	51.89%	9.66	48.11%	7.63	49.73%	6.55
desk_2	41.76%	12.95	38.92%	9.31	55.40%	7.93	76.42%	6.35
desk_3	72.77%	9.84	52.57%	7.09	52.91%	5.78	90.58%	4.60
kitchen_small_1	36.31%	7.97	34.74%	5.29	48.20%	4.12	56.81%	3.25
meeting_small_1	41.40%	3.29	38.05%	2.35	42.92%	1.74	50.63%	1.33
table_1	56.76%	4.62	38.11%	3.43	31.08%	2.49	61.49%	2.00
table_small_1	75.03%	7.50	63.30%	5.33	65.35%	3.89	83.36%	3.16
table_small_2	55.39%	9.13	45.35%	6.88	49.34%	5.04	65.88%	4.10
Average	52.77%	6.99	45.37%	5.03	49.16%	3.80	66.86%	3.02

Table 4: Performance comparison between full and a specific training set with objects from the Scenes dataset.

DeepNet	Full		Scenes	
	Acc	FPS	Acc	FPS
MobileNet v2	52.77%	6.99	67.35%	24.62
Resnet101	45.37%	5.03	61.41%	13.94
ResNeXt101 32x8d	49.16%	3.80	59.04%	8.86
EfficientNet-B7	66.86%	3.02	82.94%	5.88

and we notice that a better estimation can speed up the fine-adjustment process.

To evaluate more deeply if the ICP, after the feature-matching application, can surpass problems like a more rough estimation, we must assess an annotated pose. Unfortunately, the adopted dataset does not offer such data, and further studies may verify that affirmation on a pose-annotated dataset. However, we can evaluate the estimation correctness by employing the IoU and MIR metrics and verify if the feature-based registration step’s estimation is reliable compared to standard approaches. In Table 6 we perform such comparison regarding the AUC and FPS values of different setup of our proposed pipeline, the standard (Aldoma et al., 2012b), and the boosted (Marcon et al., 2019) pipelines.

Results of Table 6 confirm our claim that performing the object detection on the RGB images improves results compared to traditional approaches. Both standard and boosted pipelines present accuracy results worst than all trials we run in our pipeline, even considering the same conditions of descriptors and leaf size, e.g., 1 cm of leaf size in Boost $US_{0.01}$ trial. When we consider time processing, the difference is even more discrepant when our approach presents in the best case, a frame-rate of 14.18 against 0.09 FPS on the best standard approaches, which represents a remarkable improvement of more than $150\times$ in speed. When using the EfficientNet/FGR pair, our

proposal presents AUC (0.4123) three times higher than the Boosted pipeline (0.1372). We did not run the Baseline $US_{0.01}$ because this method is very time-consuming and does not represent a reasonable choice regarding the boosted version (Boost $US_{0.01}$). We found a frame rate of 0.0005 for a small set of frames experimentally. Besides, the boosted pipeline (Marcon et al., 2019) gains on accuracy and time performance regarding the traditional version, as seen on the trials with a leaf size of 0.02 (Baseline $US_{0.02}$ and Boost $US_{0.02}$), and such behavior is also expected on a smaller leaf size.

4.4.4 Time Processing Evaluation

Now we report the processing rate regarding executing the three stages of our proposed pipeline. Table 7 presents the frame processing rate based on a single target object scenario. We evaluate referring to the first stage execution (*Color only*), the early two stages (Columns *RANSAC*, and *FGR*), and a pipeline’s full execution (*+ICP*).

At first sight, one can conjecture that a RANSAC-based approach is unpromising when presenting around 2 FPS. However, considering an FGR-based process, the results are indeed encouraging, with 8 FPS for the best accurate method, and more than 13 for the others. For many applications that deal with real-time, a frame rate around eight or more is acceptable. We agree that *the facto* standard for real-time is at least 30 FPS, however, due to the modularity of our proposed pipeline, the stages are independent, and we could use the full execution only to indispensable situations.

An application scenario may include a target object’s location and pose recovering, for instance, by a robot or a visually impaired person. The system could execute a scheduled procedure, localizing this object adopting only the first stage of the pipeline, in real-

Table 5: Comparison between feature-based registration methods. Values reported consider the processing time (in seconds) for ten views of the same object and the ICP for the best one selected.

Methods	Feature-based time (\downarrow)	ICP time (\downarrow)	Inlier ratio (\uparrow)	RMSE (\downarrow)
RANSAC	0.7688	0.0061	0.2689	0.0055
FGR (Zhou et al., 2016)	0.0580	0.0075	0.1895	0.0059

Table 6: Comparison of the proposed pipeline with standard object recognition and pose estimation approaches. Baseline refer to (Aldoma et al., 2012a) and Boost to (Marcon et al., 2019). Every trial employed FPFH (Rusu et al., 2009) as local descriptor with a uniform sampling as keypoint detector. Excepting the first two rows, leaf size was set to 1 cm.

Method	AUC	FPS
Baseline $US_{0.02}$	0.0401	0.0023
Boost $US_{0.02}$	0.0868	0.0918
Boost $US_{0.01}$	0.1372	0.0339
Resnet101 + FGR	0.2228	13.8321
ResNet101 + RANSAC	0.2092	1.9649
MobileNet v2 + FGR	0.2922	13.8939
MobileNet v2 + RANSAC	0.2781	1.8905
ResNeXt101 32x8d + FGR	0.2090	14.1813
ResNeXt101 32x8d + RANSAC	0.1947	2.0268
EfficientNet-B7 + FGR	0.4123	8.9429
EfficientNet-B7 + RANSAC	0.2994	1.4344

time. Then, as the subject approaches the objective, we could execute the second stage, estimating a rough transformation, e.g., once a second. Finally, when the object is next to the user, we can run the full pipeline, including the fine-adjustment stage.

To investigate more deeply the processing time of a successfully detected object of our pipeline, we summarize how much time takes each substep in Figure 3. We can infer that two main steps negatively impact the time processing: classification and feature-based estimation. Regarding the former, the correct selection of the network to extract color features is fundamental to speed-up the whole process, presenting a significant difference between the faster (Sandler et al., 2018) and the slower (Tan and Le, 2019). We perceive a considerable impact in time processing when using RANSAC instead of FGR for the feature-based stage. In this implementation, we do not use any concurrent processing, which could significantly improve such time for both coarse pose estimation methods. Our pipeline is highly flexible, and the use of recent proposals may enhance our results on coarse estimation, for instance DGR (Choy et al., 2020).

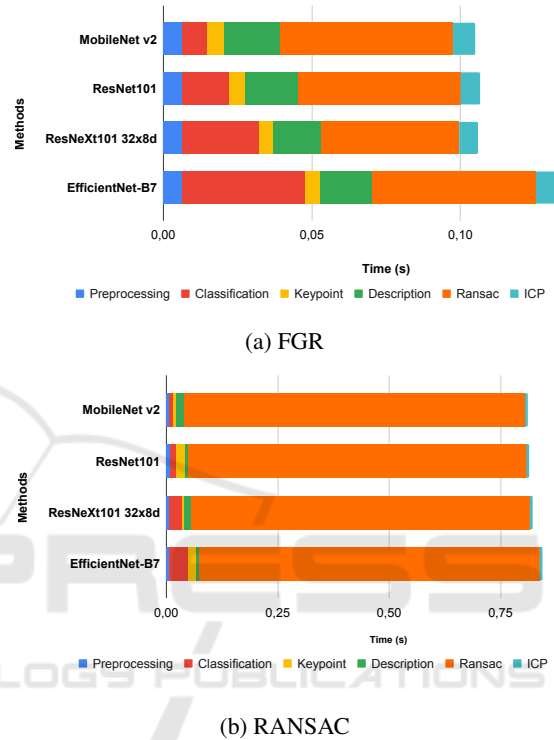


Figure 3: Processing time (in seconds) of each step of the execution of proposed approach. We consider only successfully detected objects on this comparison. (a) presents times referring to the FGR (Zhou et al., 2016) method, and (b) to RANSAC.

4.4.5 Qualitative Results

We provide qualitative visualizations of our proposed method (RANSAC + ICP) in Figure 4. Our method succeeds in aligning several different shaped models, such as planes (*cereal box*), cylinders (*soda can*, *coffee mugs*, and *flashlights*), and free form models (*caps*). As we perform a rigid transformation to align objects and scenes, the model’s choice is fundamental. Examples like the *red cap* that present a crumple on top harm the alignment estimation. Otherwise, we confirm the robustness of the combination of coarse and fine alignments on the bowl object (bottom row, on the left), partially cropped on the scene cloud. Still, our method infers the pose correctly.

Table 7: Single target pose estimation FPS. *Color only* refers to object classification, other columns refer to the pose alignment step, coarse (RANSAC and FGR) or fine (plus ICP).

	Color only	RANSAC	FGR	RANSAC + ICP	FGR + ICP
MobileNet v2 (Sandler et al., 2018)	89.49	1.89	13.89	1.82	13.57
ResNet101 (He et al., 2016)	52.45	1.96	13.83	1.81	13.39
ResNeXt101 32x8d (Xie et al., 2017)	33.73	2.03	14.18	2.09	13.32
EfficientNet-B7 (Tan and Le, 2019)	22.51	1.43	8.94	1.40	8.55

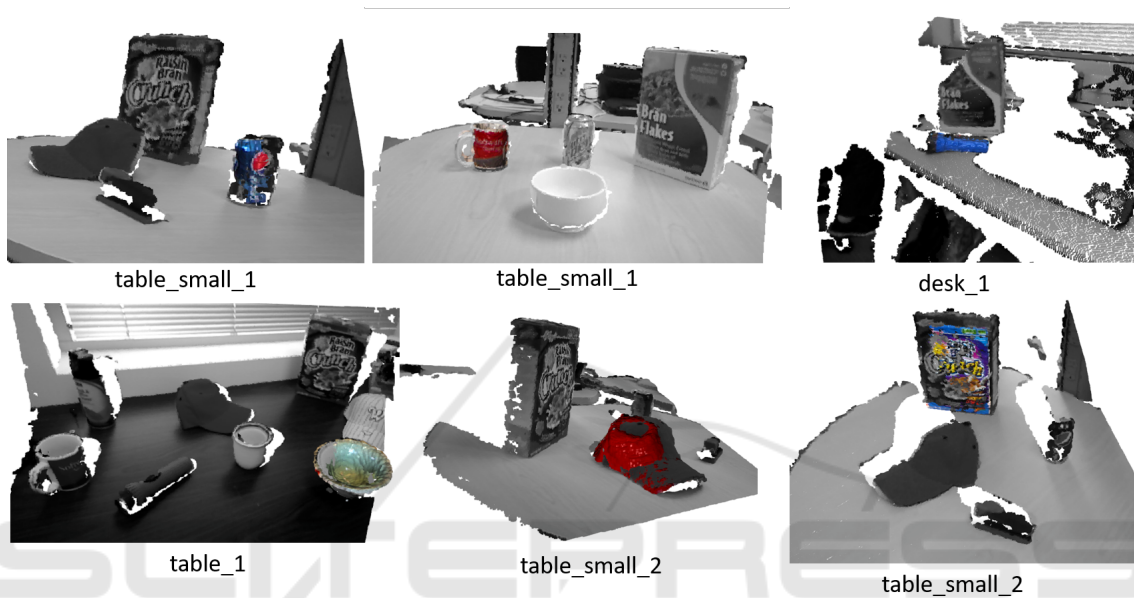


Figure 4: Qualitative visualizations of successful pose alignment.

In Figure 5, we present some wrong alignments of our proposals. We can observe that the objects' main shape weights a lot on the alignment results. For instance, the *mugs* had the body well aligned but a misalignment on the handle. We also perceive a flip on the cereal box because of the large plane at the front. The *bowl* in the rightmost example fails in aligning, though, different from the previous figure, where the method robustly handled a partial view of a *bowl*, this particular case, have about 50% only of the object visible. The ICP algorithm estimates a locally minimal transformation, and such misalignments may occur because of inaccurate inputs produced by RANSAC/FGR methods. We espy three potential solutions: using novel CNN-based estimation methods, e.g., DGR (Choy et al., 2020); adopting more robust local descriptors to the feature-based registration phase, also considering color-based approaches; increasing the number of selected 2.5D views to enhance pose covering of the scenes' objects. The last two cited solutions may negatively affect time-performance. Despite the misalignments verified, as we reduce the surface search on the scene

cloud, we always have an estimation next or even inside the 3D projection of the 2D bounding box outputted by the detection.

5 CONCLUSIONS

3D pose estimation is a challenging task, mainly for real-time applications. Sometimes developers must surrender on the precision, aiming the response time. In this paper, we introduced a novel pipeline that proposes to combine the power of color features extractors deep networks, with a local descriptors pipeline to pose estimation in point clouds. We evaluated the detection of objects and achieved almost 83% on an instance situation, in the best case. This precision is also accompanied by a high frame processing rate, arriving up to 90 FPS. The pose estimation rate is plausible for some applications, and by scheduling the stages of our pipeline, we can reach standard real-time processing. We show experimentally massive improvements concerning accuracy and time processing compared to standard approaches for ob-

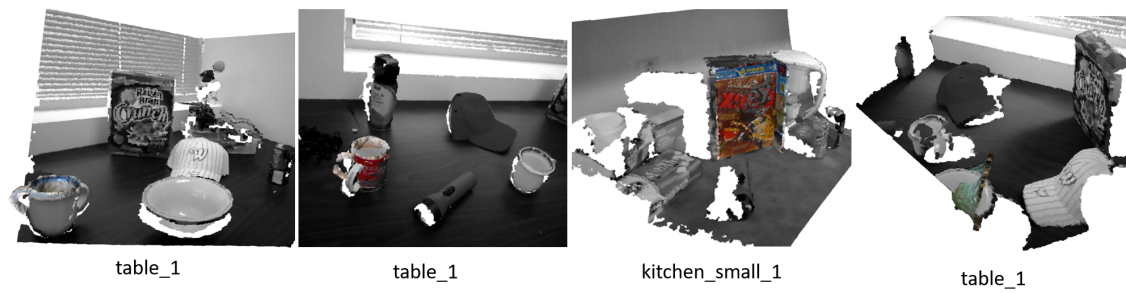


Figure 5: Qualitative visualizations of wrong pose alignment. From left to right: two examples of coffee mugs with a misoriented handles, flipped cereal box, and a rotated bowl.

ject recognition and pose estimation. Our approach is $3\times$ more efficient and $150\times$ faster than traditional and grounded methodologies.

Our three-staged detachable pipeline can be used according to the user/application needs: the color feature classification provides object detection in real-time; the feature-based registration estimates an imprecise but sometimes efficient pose of the scenes' object; the third stage performs a fine alignment of the estimation, resulting in a more accurate result. We believe that our proposal's adoption may help researchers and the industry develop reliable and time-efficient solutions for scene recognition problems from RGB-D data.

Parallelization strategies can improve time results even more and also different local descriptors and keypoint extractors could support this. Findings on the deepnets architectures can help developing an integrated region proposal and object detection algorithm, and state-of-the-art deep learning methods such as SSD (Liu et al., 2016), YOLO (Redmon et al., 2016), and EfficientDet (Tan et al., 2020) enable such potentiality.

ACKNOWLEDGMENTS

We would like to thank UTFPR-DV for partly supporting this research work.

REFERENCES

- Agrawal, P., Girshick, R., and Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pages 329–344. Springer.
- Aldoma, A., Marton, Z., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., and Vincze, M. (2012a). Three-dimensional object recognition and 6 DoF pose estimation. *IEEE Robotics & Automation Magazine*, pages 80–91.
- Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S., and Vincze, M. (2012b). Tutorial: Point cloud library: Three-dimensional object recognition and 6 DoF pose estimation. *IEEE Robotics & Automation Magazine*, 19(3):80–91.
- Asif, U., Bennamoun, M., and Sohel, F. A. (2017). A multi-modal, discriminative and spatially invariant CNN for RGB-D object labeling. *IEEE transactions on pattern analysis and machine intelligence*, 40(9):2051–2065.
- Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics.
- Bo, L., Ren, X., and Fox, D. (2013). Unsupervised feature learning for RGB-D based object recognition. In *Experimental robotics*, pages 387–402. Springer.
- Caglayan, A., Imamoglu, N., Can, A. B., and Nakamura, R. (2020). When CNNs meet random RNNs: Towards multi-level analysis for RGB-D object and scene recognition. *arXiv preprint arXiv:2004.12349*.
- Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155.
- Choi, S., Zhou, Q.-Y., and Koltun, V. (2015). Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565.
- Choy, C., Dong, W., and Koltun, V. (2020). Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., and Kwok, N. M. (2016). A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

- Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al. (2018). Bop: Benchmark for 6D object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34.
- Hou, Q., Cheng, M.-M., Hu, X., Borji, A., Tu, Z., and Torr, P. H. (2017). Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Lai, K., Bo, L., Ren, X., and Fox, D. (2011a). A large-scale hierarchical multi-view RGB-D object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE.
- Lai, K., Bo, L., Ren, X., and Fox, D. (2011b). Sparse distance learning for object recognition combining rgb and depth information. In *2011 IEEE International Conference on Robotics and Automation*, pages 4007–4013. IEEE.
- Liu, H., Li, F., Xu, X., and Sun, F. (2018). Multi-modal local receptive field extreme learning machine for object recognition. *Neurocomputing*, 277:4–11.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- Marcon, M., Spezialetti, R., Salti, S., Silva, L., and Di Stefano, L. (2019). Boosting object recognition in point clouds by saliency detection. In *International Conference on Image Analysis and Processing*, pages 321–331. Springer.
- Ouadiay, F. Z., Zrira, N., Bouyakhf, E. H., and Himmi, M. M. (2016). 3d object categorization and recognition based on deep belief networks and point clouds. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO.*, pages 311–318. INSTICC, SciTePress.
- Park, J., Zhou, Q.-Y., and Koltun, V. (2017). Colored point cloud registration revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 143–152.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE.
- Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2008). Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE.
- Salti, S., Tombari, F., and Di Stefano, L. (2014). SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.
- Schwarz, M., Schulz, H., and Behnke, S. (2015). RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1329–1335. IEEE.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114.
- Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790.
- Vock, R., Dieckmann, A., Ochmann, S., and Klein, R. (2019). Fast template matching and pose estimation in 3D point clouds. *Computers & Graphics*, 79:36–45.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.
- Zaki, H. F., Shafait, F., and Mian, A. (2019). Viewpoint invariant semantic object and scene categorization with RGB-D sensors. *Autonomous Robots*, 43(4):1005–1022.
- Zhou, Q.-Y., Park, J., and Koltun, V. (2016). Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer.
- Zia, S., Yuksel, B., Yuret, D., and Yemez, Y. (2017). RGB-D object recognition using deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 896–903.