Review article

A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators☆

Caner Sahin *, Guillermo Garcia-Hernando, Juil Sock, Tae-Kyun Kim

Imperial College London, UK

ARTICLE INFO

ABSTRACT

Object pose recovery has gained increasing attention in the computer vision field as it has become an important problem in rapidly evolving technological areas related to autonomous driving, robotics, and augmented reality. Existing review-related studies have addressed the problem at visual level in 2D, going through the methods which produce 2D bounding boxes of objects of interest in RGB images. The 2D search space is enlarged either using the geometry information available in the 3D space along with RGB (Mono/Stereo) images, or utilizing depth data from LIDAR sensors and/or RGB-D cameras. 3D bounding box detectors, producing category-level amodal 3D bounding boxes, are evaluated on gravity aligned images, while full 6D object pose estimators are mostly tested at instance-level on the images where the alignment constraint is removed. Recently, 6D object pose estimation is tackled at the level of categories. In this paper, we present the first comprehensive and most recent review of the methods on object pose recovery, from 3D bounding box detectors to full 6D pose estimators. The methods mathematically model the problem as a classification, regression, classification & regression, template matching, and point-pair feature matching task. Based on this, a mathematical-model-based categorization of the methods is established. Datasets used for evaluating the methods are investigated with respect to the challenges, and evaluation metrics are studied. Quantitative results of experiments in the literature are analyzed to show which category of methods best performs across what types of challenges. The analyses are further extended comparing two methods, which are our own implementations, so that the outcomes from the public results are further solidified. Current position of the field is summarized regarding object pose recovery, and possible research directions are identified.

© 2020 Elsevier B.V. All rights reserved.

Contents

1. Introduction

Object pose recovery is an important problem in the computer vision field, which, to the full extent, determines the 3D position and 3D rotation of an object in camera-centered coordinates. It has extensively been studied in the past decade as it is of great importance to many rapidly evolving technological areas such as autonomous driving, robotics, and augmented reality.

Autonomous driving, as being a focus of attention of both industry and research community in recent years, fundamentally requires accurate object detection and pose estimation in order for a vehicle to avoid collisions with pedestrians, cyclists, and cars. To this end, autonomous vehicles are equipped with active LIDAR sensors [142,146], passive Mono/Stereo (Mo/St) RGB/D/RGB-D cameras [84,87], and their fused systems [82,108].

Robotics has various sub-fields that vastly benefit from accurate object detection and pose estimation. One prominent field is robotic manipulation, in which industrial parts are grasped and are placed to a desired location [62,56,57,60,91,202,200]. Amazon Picking Challenge (APC) [1] is an important example demonstrating how object detection and 6D pose estimation, when successfully performed, improves the autonomy of the manipulation, regarding the automated handling of parts by robots [55,58,59]. Household robotics is another field where the ability of recognizing objects and accurately estimating their poses is a key element [72,97–99]. This capability is needed for such robots, since they should be able to navigate in unconstrained human environments calculating grasping and avoidance strategies. Autonomous flight performance of Unmanned Aerial Vehicles (UAVs) is improved integrating object pose recovery into their control systems. Successful results are particularly observed in environments where GPS information is denied or is inaccurate [54,61,53].

Simultaneous Localization and Mapping (SLAM) is the process of building the map of an unknown environment and determining the location of a robot concurrently using this map. Recently introduced SLAM algorithms have object-oriented design [63,64,65,66,67,68,69,70,71]. Accurate object pose parameters provide camera-object constraints and lead to better performance on localization and mapping.

Augmented Reality (AR) technology transforms the real world into an interactive and digitally manipulated space, superimposing computer generated content onto the real world. Augmentation of reality requires the followings: (i) a region to augment must be detected in the image (object detection) and (ii) the rotation for the augmented content must be estimated (pose estimation) [45]. Virtual Reality (VR) transports users into a number of real-world and imagined environments [46,49,50,52], and Mixed Reality (MR) [47,48,51] merges both AR and VR technologies. Despite slight differences, their success lies on accurate detection and pose estimation of objects.

In the progress of our research on object pose recovery, we have come across several review-related publications [8,9,10,11,12,13,109,199] handling object pose recovery at visual level in 2D. These publications discuss the effects of challenges, such as occlusion, clutter, texture, etc., on the performances of the methods, which are mainly evaluated on large-scale datasets, e.g., ImageNet [6], PASCAL [7]. In [8], the effect of different

context sources, such as geographic context, object spatial support, on 2D object detection is examined. Hoiem et al. [9] evaluate the performances of several baselines on the PASCAL dataset particularly analyzing the reasons why false positives are hypothesized. Torralba et al. [11] compare several datasets regarding the involved samples, cross-dataset generalization, and relative data bias. Since there are less number of object categories in PASCAL, Russakovsky et al. [10] use ImageNet in order to do meta-analysis and to examine the influences of color, texture, etc., on the performances of object detectors. The retrospective evaluation [12] and benchmark [13] studies perform the most comprehensive analyses on 2D object localization and category detection, by examining the PASCAL Visual Object Classes (VOC) Challenge, and the ImageNet Large Scale Visual Recognition Challenge, respectively. Recently proposed study [109] systematically reviews deep learning based object detection frameworks including several specific tasks, such as salient object detection, face detection, and pedestrian detection. These studies introduce important implications for generalized object detection, however, the reviewed methods and discussions are restricted to visual level in 2D.

2D-driven 3D bounding box (BB) detection methods enlarge the 2D search space using the available appearance and geometry information in the 3D space along with RGB images [90,151,153,154,144]. The methods presented in [84,155] directly detect 3D BBs of the objects in a monocular RGB image exploiting contextual models as well as semantics. Apart from [84,155], the methods which directly detect 3D BBs use stereo images [92], RGB-D cameras [86,93,94,95,152], and LIDAR sensors [152] as input. There are also several methods fusing multiple sensor inputs, and generating 3D BB hypotheses [107]. Without depending on the input (RGB (Mo/St), RGB-D, or LIDAR), 3D BB detection methods produce **oriented** 3D BBs, which are parameterized with center $\mathbf{x} = (x,y,z)$, size $\mathbf{d} = (d_w, d_h, d_l)$, and orientation ($\theta_y$) around the gravity direction. Note that, any 3D BB detector can be extended to the 6D space, however, available methods in the literature are mainly evaluated on the KITTI [110], NYU-Depth v2 [111], and SUN RGB-D [112] datasets, where the objects of interest are aligned with the gravity direction. As required by the datasets, these methods work at the level of categories e.g., cars, bicycles, pedestrians, chairs, beds, producing **amodal** 3D BBs, i.e., the minimum 3D BB enclosing the object of interest.

The search space of the methods engineered for object pose recovery is further enlarged to 6D [165,171,175,183], i.e., 3D translation $\mathbf{x} = (x,y,z)$ and 3D rotation $\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$. Some of the methods estimate 6D poses of objects of interest in RGB-D images. The holistic template matching approach, Linemod [2], estimates cluttered object's 6D pose using color gradients and surface normals. It is improved by discriminative learning in [24]. FDCM [23] is used in robotics applications. Drost et al. [17] create a global model description based on oriented point pair features and match that model locally using a fast voting scheme. It is further improved in [43] making the method more robust across clutter and sensor noise. Latent-class Hough forests (LCHF) [4,26], employing one-class learning, utilize surface normal and color gradient features in a part-based approach in order to provide robustness across occlusion. Occlusion-aware features [27] are further formulated. The studies in [28,29] cope with texture-less objects. More recently, feature representations are learnt in an unsupervised fashion using deep

convolutional (conv) networks (net) [35,36]. While these methods fuse data coming from RGB and depth channels, a local belief propagation based approach [73] and an iterative refinement architecture [31,32] are proposed in depth modality [74]. 6D object pose estimation is recently achieved from RGB only [164,170,172,173,174,30,37,38,40], and the current paradigm is to adopt CNNs [157,158,169]. BB8 [40] and Tekin et al. [38] perform corner-point regression followed by PnP. Typically employed is a computationally expensive post processing step such as iterative closest point (ICP) or a verification net [37]. As mainly being evaluated on the LINEMOD [2], OCCLUSION [28], LCHF [4], and T-LESS [42] datasets, full 6D object pose estimation methods typically work at the level of **instances**. However, recently proposed methods in [113,114] address the 6D object pose estimation problem at **category level**, handling the challenges, *e.g.*, distribution shifts, intra-class variations [206].

In this study, we present a comprehensive review on object pose recovery, reviewing the methods from 3D BB detectors to full 6D pose estimators. The reviewed methods mathematically model object pose recovery as a classification, regression, classification & regression, template matching, and point-pair feature matching task. Based on this, a mathematical-model-based categorization of the methods is established, and 5 different categories are formed. Advances and drawbacks are further studied in order to eliminate ambiguities in between the categories. Each individual dataset in the literature involves some kinds of challenges, across which one can measure the performance by testing a method on it. The challenges of the problem, such as viewpoint variability, occlusion, clutter, intra-class variation, and distribution shift are identified by investigating the datasets created for object pose recovery. The protocols used to evaluate the methods' performance are additionally examined. Once we introduce the categorization of the reviewed methods and identify the challenges of the problem, we next reveal the performance of the methods across the challenges. To this end, publicly available recall values, which are computed under uniform scoring criteria of the Average Distance (AD) metric [2], are analyzed. The analyses are further extended comparing two more methods [2,4], which are our own implementations, using the Visible Surface Discrepancy (VSD) protocol [3] in addition to AD. This extension mainly aims to leverage the outcomes obtained from the public results and to complete the discussions on the problem linking all its aspects. The current position of the field is lastly summarized regarding object pose recovery, and possible research directions are identified.

Benchmark for 6D object pose estimation (BOP) [5], contributes a unified format of datasets, an on-line evaluation system based on the VSD metric, and a comprehensive evaluation of 15 different methods [205]. The benchmark addresses object pose recovery only considering 6D object pose estimation methods, which work at instance-level. The analyses on methods' performance are relatively limited to the technical background of the methods evaluated. In this study, we contribute the most comprehensive review of the methods on object pose recovery, from 3D BB detectors to full 6D pose estimators, which work both at instance- and category-level. Our analyses on quantitative results interpret the reasons behind methods' strength and weakness with respect to the challenges.

1.1. Contributions

To the best of our knowledge, this is the first comprehensive and most recent study reviewing the methods on object pose recovery. Our contributions are as follows:

- A mathematical-model-based categorization of object pose estimation methods is established. The methods concerned range from 3D BB detectors to full 6D pose estimators.
- Datasets for object pose recovery are investigated in order to identify

its challenges.
- Publicly available results are analyzed to measure the performance of object pose estimation methods across the challenges. The analyses are further solidified by comparing in-house implemented methods.
- Current position of the field on object pose recovery is summarized to unveil existing issues. Furthermore, open problems are discussed to identify potential future research directions.

1.2. Organization of the review

The paper is organized as follows: Section 2 formulates the problem of object pose recovery at instance- and category-level. Section 3 presents the categorization of the reviewed methods. In the categorization, there are five categories of the methods: classification, regression, classification & regression, template matching, and point-pair feature matching. Section 4 depicts the investigations on the datasets and examines the evaluation protocols, and Section 5 shows the analyses on public results and the results of in-house implemented methods. Section 6 summarizes the current position of the field and concludes the study.

1.3. Denotation

Throughout the paper, we denote scalar variables by lowercase letters (*e.g.*, x) and vector variables bold lowercase letters (*e.g.*, $\mathbf{x}$). Capital letters (*e.g.*, P) are adopted for specific functions or variables. We use bold capital letters (*e.g.*, $\mathbf{I}$) to denote a matrix or a set of vectors. Table 1 lists the symbols utilized throughout this review. Except the symbols in the table, there may be some symbols for a specific reference. As these symbols are not commonly employed, they are not listed in the table but will be rather defined in the context.

2. Object pose recovery

In this study, we present a review of the methods on object pose recovery, from 3D BB detectors to full 6D pose estimators. The methods deal with *seen* objects at instance-level and *unseen* objects at the level of categories. In this section, we firstly formulate object pose recovery for instance-level. Given an RGB/D/RGB-D image $\mathbf{I}$ where a seen instance S of the interested object O exists, object pose recovery is cast as a joint probability estimation problem and is formulated as given below:

$$(\mathbf{x}, \mathbf{\Theta})^* = \arg \max_{\mathbf{x}, \mathbf{\Theta}} P(\mathbf{x}, \mathbf{\Theta} | \mathbf{I}, S, O) \tag{1}$$

where $\mathbf{x} = (x, y, z)$ is the 3D translation and $\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$ is the 3D rotation of the instance S. $(\theta_r, \theta_p, \theta_y)$ depicts the Euler angles, roll, pitch, and yaw, respectively. According to Eq. (1), instance-level methods of object pose recovery target to maximize the joint posterior density of the 3D translation $\mathbf{x}$ and 3D rotation $\mathbf{\Theta}$. When the image $\mathbf{I}$ involves multiple instances $\mathcal{S} = \{S_i | i = 1, ..., n\}$ of the object of interest, the problem formulation becomes:

$$(\mathbf{x}_i, \mathbf{\Theta}_i)^* = \arg \max_{\mathbf{x}_i, \mathbf{\Theta}_i} P(\mathbf{x}_i, \mathbf{\Theta}_i | \mathbf{I}, \mathcal{S}, O), i = 1, ..., n. \tag{2}$$

Given an unseen instance U of a category of interest C, the object pose recovery problem is formulated at the level of categories by transforming Eq. (1) into the following form:

$$(\mathbf{x}, \mathbf{\Theta})^* = \arg \max_{\mathbf{x}, \mathbf{\Theta}} P(\mathbf{x}, \mathbf{\Theta} | \mathbf{I}, U, C). \tag{3}$$

Table 1
Denotations employed in this review.

Symbol	Description	Symbol	Description
$\mathbf{I}$	Input image	$\mathbf{x} = (x, y, z)$	3D translation
P	Probability	$\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$	3D rotation
O	Object	$\mathbf{d} = (d_w, d_h, d_l)$	3D BB dimension
S	Seen instance	$\mathbf{p}_{3D} = (p_x, p_y, p_z)$	3D BB corner position
$\mathcal{S}$	Set (seen inst)	$\mathbf{p}_{3D}^{proj} = (p_x, p_y)$	Projection of $\mathbf{p}_{3D}$
U	Unseen instance	$\mathbf{p}_{2D} = (p_x, p_y)$	2D BB corner position
$\mathcal{U}$	Set (unseen inst)	$\theta_r, \theta_p, \theta_y$	Roll, pitch, yaw
C	Category	d_w, d_h, d_l	Width, height, length
M	3D model	i, j, k, n, c	General index
IG	Information gain	$\mathbf{p}_{2D_c}$	2D BB center
IoU_{3D}	3D IoU loss	θ_{cvp}	Camera viewpoint
L_{hinge}	Hinge loss	θ_{ip}	In-plane rotation
L_{pair_d}	Dynamic pair loss	L_{log}	log-loss
L_{pose}	Pose loss	L_{1_s}	Smooth $L1$ loss
L_{pair}	Pair loss	L_{ce}	Cross entropy
L_{tri_d}	Dynamic triplet loss	L_{il}	Insensitive loss
$L_{0/1}$	0/1 loss	L_2	L2 loss
L_{tri}	Triplet loss	L_1	L1 loss

When the image $\mathbf{I}$ involves multiple unseen instances $\mathcal{U} = \{U_i | i = 1, ..., m\}$ of the category of interest C, Eq. (2) takes the following form:

$$(\mathbf{x}_i, \mathbf{\Theta}_i)^* = \arg \max_{\mathbf{x}_i, \mathbf{\Theta}_i} P(\mathbf{x}_i, \mathbf{\Theta}_i | \mathbf{I}, \mathcal{U}, C), i = 1, ..., m. \qquad (4)$$

Based on this formulation, any 3D BB detection method evaluated on any dataset of categories, e.g., KITTI [110], NYU-Depth v2 [111], SUN RGB-D [112], searches for the accurate 3D translation $\mathbf{x}$ and the rotation angle around the gravity direction θ_y of 3D BBs, which are also the ground truth annotations. Unlike 3D BB detectors, full 6D object pose estimators, as being evaluated on the LINEMOD [2], OCCLUSION [28], LCHF [4], and T-LESS [42] datasets, search for the accurate full 6D poses, 3D translation $\mathbf{x} = (x, y, z)$ and 3D rotation $\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$, of the objects of interest.

3. Methods on object pose recovery

Any method for the problem of object pose recovery estimates the object pose parameters, 3D translation $\mathbf{x} = (x, y, z)$ **and/or** 3D rotation $\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$. We categorize the methods reviewed in this paper based on their problem modeling approaches: The methods which estimate the pose parameters formulating the problem as a classification task are involved within the *classification*, while the ones regressing the parameters are included in the *regression*. *Classification & regression* combines both classification and regression tasks to estimate the objects' 3D translation and/or 3D rotation. *Template matching* methods estimate the objects' pose parameters matching the annotated and represented templates in the feature space. *Point-pair feature matching* methods use the relationships between any two points, such as distance and angle between normals to represent point-pairs. Together with a hash table and an efficient voting scheme, the pose parameters of the target objects are predicted. These 5 different modeling approaches, classification, regression, classification & regression, template matching, point-pair feature matching, form a **discriminative** categorization allowing us to review the state-of-the-art methods on the problem of object pose recovery, from 3D to 6D.

When we review the methods, we encounter several 3D detection methods, which are 2D-driven. 2D-driven 3D methods utilize any off-the-shelf 2D detectors e.g., R-CNN [115], Fast R-CNN [116], Faster R-CNN [117], R-FCN [118], FPN [119], YOLO [120], SSD [121], GOP [122], or MS-CNN [123], to firstly detect the 2D BBs of the objects of interest [156], which are then lifted to 3D space, and hence making their performance dependent on the 2D detectors. Besides, several 3D detectors and full 6D pose estimators, which directly generate objects' 3D translation and/or 3D rotation parameters, are built on top of those. Decision

forests [127] are important for the problem of object pose recovery. Before delving into our categorization of the methods, we first briefly mention several 2D detectors and decision forests.

R-CNN [115], from an input RGB image, generates a bunch of arbitrary size of region proposals employing a selective search scheme [124], which relies on bottom-up grouping and saliency cues. Region proposals are warped into a fixed resolution and are fed into the CNN architecture in [125] to be represented with CNN feature vectors. For each feature vector, category-specific SVMs generate a score whose region is further adjusted with BB regression, and non-maximum suppression (NMS) filtering. The 3 stages of process allow R-CNN to achieve approximately 30% improvement over previous best 2D methods, e.g., DPM HSC [126].

Fast R-CNN [116] takes an RGB image and a set of region proposals as input. The whole image is processed to produce a conv feature map, from which a fixed-length feature vector is generated for each region proposal. Each feature vector is fed into a set of Fully Connected (FC) layers, which eventually branching into softmax classification and BB regression outlets. Each training Region of Interest (RoI) is labeled with a ground-truth class and a ground-truth regression target. Fast R-CNN introduces a multi-task loss on each RoI to jointly train for classification and BB regression.

Faster R-CNN [117] presents Region Proposal Networks (RPN), which takes an RGB image as input and predicts rectangular object proposals along with their objectness score. As being modeled with a fully conv net, RPN shares computation with a Fast R-CNN. Sliding over the conv feature map, it operates on the last conv layer, with the preceding layers shared with the object detection net. It is fully connected to a $n \times n$ spatial window of the conv feature map. In each sliding window, a low dimensional feature vector is generated for each of the k anchors, which is then fed into both box classification and box regression layers.

YOLO [120], formulating the detection problem as a regression task, predicts BBs and class probabilities directly from full images. The whole process is conducted in a single net architecture, which can be trained in an end-to-end fashion. The architecture divides the input image into an $n \times n$ grid, each cell of which is responsible for predicting the object centered in that grid cell. Each cell predicts k BBs along with their confidence scores. The confidence is defined as the multiplication of $P(O)$ and IOU_{pred}^{truth}, resulting $P(O) * IOU_{pred}^{truth}$, which indicates how likely there object exists ($P(O) \geq 0$). Each grid cell also calculates $P(C|O)$, conditional class probabilities conditioned on the grid cell containing an object. Class specific confidence scores are obtained during testing, multiplying the conditional class probabilities and the individual box confidence predictions.

SSD [121] architecture, taking an RGB image as input, produces 2D BBs and confidences of the objects of a given category, which are then filtered by the NMS filtering to produce the final detections. The architecture discretizes the output space of the BBs, utilizing a set of default anchor boxes which have different aspect ratios and scales.

Decision Forests [127,134] are ensemble of randomized decision trees. Each leaf node of each decision tree of a trained decision forest stores a prediction function. A random decision tree is trained independently from the other trees, and it can be regarded as a weak classifier. In the test, any input ending up at a leaf node, is associated with the stored prediction function. The outcome coming from each tree is averaged, since many weak classifiers, if combined, produce more accurate results [128,129,130,131,132]. In classification forests, information gain is often used as the quality function Q_{cla} and in regression tasks, the training objective Q_{reg} is to minimize the variance in translation offset vectors and rotation parameters. For pose regression problems, Hough voting process [133] is usually employed.

3.1. Classification

Overall schematic representation of the classification-based methods is shown in Fig. 1. In the figure, the blocks drawn with
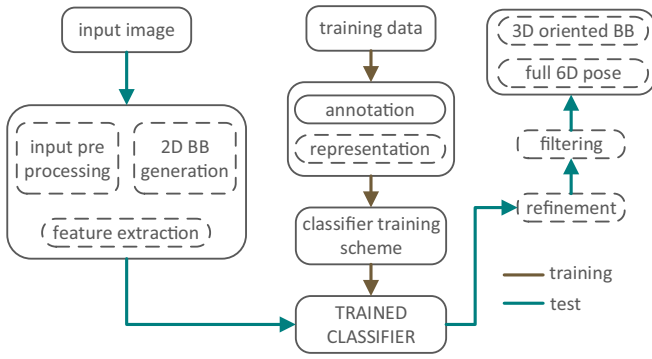
Fig. 1. Overall schematic representation of classification-based methods: blocks drawn with continuous lines are employed by all methods while dashed-line blocks are operated by clusters of specific methods (see the text and Table 2 for details).

continuous lines are employed by all methods, and depending on the architecture design, dashed-line blocks are additionally operated by the clusters of specific methods.

3.1.1. Training phase

During an off-line stage, classifiers are trained based on synthetic or real data. Synthetic data are generated using the 3D model M of an interested object O, and a set of RGB/D/RGB-D images are rendered from different camera viewpoints. The 3D model M can either be a CAD or a reconstructed model, and the following factors are considered when deciding the size of the data:

- Reasonable viewpoint coverage. In order to capture reasonable viewpoint coverage of the target object, synthetic images are rendered by placing a virtual camera at each vertex of a subdivided icosahedron of a fixed radius. The hemisphere or full sphere of icosahedron can be used regarding the scenario.
- Object distance. Synthetic images are rendered at different scales depending on the range in which the target object is located.

Computer graphic systems provide precise data annotation, and hence synthetic data generated by these systems are used by the classification-based methods [88,81,89,27,28,29,33]. It is hard to get accurate object pose annotations for real images, however, there are classification-based methods using real training data [28,29,33,77,88,89,96,104,149]. Training data are annotated with pose

parameters *i. e.*, 3D translation $\mathbf{x} = (x,y,z)$, 3D rotation $\mathbf{\Theta} = (\theta_r, \theta_p, \theta_y)$, or both. Once the training data are generated, the classifiers are trained using related loss functions.

3.1.2. Testing phase

A real test image, during an on-line stage, is taken as input by the classifiers. 2D-driven 3D methods [104,88,81] first extract a 2D BB around the object of interest (2D BB generation block), which is then lifted to 3D. Depending on the input, the methods in [88,81,89,149,96,77,27] employ a pre-processing step on the input image and then generate 3D hypotheses (input pre-processing block). 6D object pose estimators [27,28,29,33] extract features from the input images (feature extraction block), and using the trained classifiers, estimate objects' 6D pose. Several methods further refine the output of the trained classifiers [104,81,149,28,29,33] (refinement block), and finally hypothesize the object pose after filtering.

Table 2 details the classification-based methods. GS3D [104] concentrates on extracting the 3D information hidden in a 2D image to generate accurate 3D BB hypotheses. It modifies Faster R-CNN [117] to classify the rotation θ_y in RGB images in addition to the 2D BB parameters. Utilizing another CNN architecture, it refines the object's pose parameters further classifying $\mathbf{x}$, $\mathbf{d}$, and θ_y. Papon et al. [88] estimate semantic poses of common furniture classes in complex cluttered scenes. The input is converted into the intensity (RGB) & surface normal (D) image. 2D BB proposals are generated using the 2D GOP detector [122], which is then lifted to 3D space further classifying θ_y and z using the bin-based cross entropy loss L_{ce}. Gupta et al. [81] start with the segmented region output from [76], the depth channel of which is further processed to acquire surface normal image. The rotation θ_y of the segmented region is classified by the L_{ce} loss. The method further refines the estimated coarse pose employing ICP over objects' CAD models. The input depth image of the SVM-based Sliding Shapes (SS) [89] is pre-processed to obtain voxelized point cloud. The classifier is trained using the hinge loss L_{hinge} to classify the parameters $\mathbf{x}$, $\mathbf{d}$, and θ_y. Ren et al. [149] propose Cloud of Oriented Gradients (COG) descriptors which link 2D appearance to 3D point cloud. The method produces 3D box proposals, the parameters $\mathbf{x}$, $\mathbf{d}$, and θ_y of which are after NMS filtering further refined by another classifier (s-SVM) to obtain the final detections. The sparse nature of point clouds is leveraged in Wang et al. [96] employing a voting scheme which enables a search over all putative object locations. It estimates the parameter $\mathbf{x}$ by using an s-SVM after converting the LIDAR image into 3D grid. As conducted in Wang et al. [96], the input LIDAR image of Vote3Deep [77] is pre-

Table 2
Classification-based methods (CNN: **C**onvolutional **N**eural **N**etwork, CRF: **C**onditional **R**andom **F**ield, ICP: **I**terative **C**losest **P**oint, NMS: **N**on-**M**aximum **S**uppression, R: **R**eal data, RF: **R**andom **F**orest, S: **S**ynthetic data, s-SVM: **s**tructured-**S**upport **V**ector **M**achine, for symbols see Table 1).

Method	Input	Input pre-processing	Training data	Classification parameters	Classifier training	Trained classifier	Refinement step	Filtering	Level
2D-DRIVEN 3D									
GS3D [104]	RGB	✗	R	θ_y	L_{ce}	CNN	CNN	✗	Category
Refinement step	RGB	✗	R	$\mathbf{x}$, $\mathbf{d}$, θ_y	L_{ce}	CNN	✗	✗	
Papon et al. [88]	RGB-D	Intensity & normal	R & S	θ_y, z	L_{ce}	CNN	✗	NMS	Category
Gupta et al. [81]	RGB-D	Normal	S	θ_y	L_{ce}	CNN	ICP	✗	Category
3D									
Sliding Shapes [89]	Depth	3D grid	R & S	$\mathbf{x}$, $\mathbf{d}$, θ_y	L_{hinge}	SVM	✗	NMS	Category
Ren et al. [149]	RGB-D	3D grid	R	$\mathbf{x}$, $\mathbf{d}$, θ_y	IoU_{3D}	s-SVM	s-SVM	NMS	Category
Refinement step	RGB-D	3D grid	R	$\mathbf{x}$, $\mathbf{d}$, θ_y	IoU_{3D}	s-SVM	✗	✗	
Wang et al. [96]	LIDAR	3D grid	R	$\mathbf{x}$	L_{hinge}	SVM	✗	NMS	Category
Vote3Deep [77]	LIDAR	3D grid	R	$\mathbf{x}$, θ_y	L_{hinge}	CNN	✗	NMS	Category
6D									
Bonde et al. [27]	Depth	3D grid	S	$\mathbf{\Theta}$	IG	RF	✗	✗	Instance
Brachmann et al. [28]	RGB-D	✗	R & S	$\mathbf{x}$	IG	RF	ICP	✗	Instance
Krull et al. [29]	RGB-D	✗	R & S	$\mathbf{x}$	IG	RF	CNN	✗	Instance
Refinement step	Depth	✗	R & S	$\mathbf{x}$, $\mathbf{\Theta}$	Log-like	CNN	✗	✗	
Michel et al. [33]	RGB-D	✗	R & S	$\mathbf{x}$	IG	RF	CRF & ICP	✗	Instance
Refinement step	RGB-D	✗	R & S	$\mathbf{x}$, $\mathbf{\Theta}$	✗	CRF	ICP	✗	

processed to acquire sparse 3D grid, which is then fed into the CNN architecture to classify $\mathbf{x}$ and θ_y parameters.

The search space of the 3D BB detectors is further enlarged to 6D [27,28,29,33]. The random forest-based method of Bonde et al. [27] estimates objects' 6D pose in depth images in the simultaneous presence of occlusion, clutter, and similar-looking distractors. It is trained based on the information gain IG to classify $\mathbf{\Theta}$. Despite the fact that this method classifies 3D rotation $\mathbf{\Theta}$, after extracting *edgelet* features, it voxelizes the scene and selects the center of the voxel, which has the minimum distance between edgelet features, as the center of the detected object. Brachmann et al. [28] introduce a random forest-based method for the 6D object pose estimation problem. Unlike Bonde et al. [27], this method classifies 3D translation parameters $\mathbf{x}$ of the objects of interest, the exact 6D poses of which are obtained using ICP-variant algorithm. Based upon Brachmann et al. [28], Krull et al. [29] and Michel et al. [33] present novel RGB-D-based methods for 6D object pose recovery. The main contributions of [29] and [33] are on the refinement step. Krull et al. [29] train a novel CNN architecture which learns to compare observation and renderings, and Michel et al. [33] engineer a novel Conditional Random Field (CRF) for generating a pool of pose hypotheses.

The training phase of the methods [104,149,96,77] are based on real data, while the ones in [81,27] are trained using synthetic images. Papon et al. [88] train the method based on synthetic images, however they additionally utilize real positive data to fine-tune the classifier, a heuristic that improves the accuracy of the method across unseen instances. Sliding Shapes [89] uses positive synthetic and negative real images to train the SVMs. Brachmann et al. [28], and accordingly Krull et al. [29] and Michel et al. [33] train the forests based on positive and negative real data, and positive synthetic data. Amongst the methods, the hypotheses of the ones presented in [88,89,149,96,77] are NMS-filtered. All 2D-driven 3D methods and 3D methods work at the level of categories. Full 6D pose estimators are designed for instance-level object pose estimation.

3.2. Regression

Fig. 2 demonstrates the overall schematic representation of the regression-based methods. The training and the testing phases of the regression-based methods are similar to that of the classification-based methods. During an off-line stage, regressors are trained based on real or synthetic annotated data using the related loss functions. In an on-line stage, a real image is taken as input by any of the regressors. 2D-driven 3D methods [101,85,83,147] first extract a 2D BB around the object of interest (2D BB generation block), which is then lifted to 3D. Input pre-processing (input pre-processing block) is conducted by several methods [101,79,150,142,146,80,82,143,108] to make the input data suitable for the trained regressor. 6D object pose estimators [40,38,163,167,168,159,160,31,32,4,35,161] extract features from the input images, and using the trained regressor, estimate objects' 6D pose. Several methods further refine the output of the trained regressors [101,83,79,82,108,40,38,163,167,168,159,160,31,32,4,35,161] (refinement block), and finally hypothesize the object pose after filtering.

Table 3 elaborates the regression-based methods. Wang et al. [101] argue that the performance gap in between the LIDAR-based and the RGB-based methods is not the quality of the data, but the representation. They convert image-based depth maps acquired from RGB (Mo/St) to pseudo-LiDAR (p-LIDAR) representation, which is then fed as input to the regression-based detection algorithm AVOD [108] (This representation is also given as input to FrustumPNet [100] and will be discussed in the Classification & regression subsection). The RGB-D-based method of Deng et al. [85] first detects 2D BBs using Multiscale Combinatorial Grouping (MCG) [76,138,139]. Engineering a Fast R-CNN-based architecture, it regresses $\mathbf{x}$, $\mathbf{d}$, and θ_y. Another RGB-D-based 2D-driven 3D method [83] generates 2D BB hypotheses using Faster R-CNN, and then regresses $\mathbf{d}$ exploiting a Multi-Layer Perceptron (MLP), whose training

is based on L_{1s}. Unlike [85], the hypotheses produced by MLP are further refined utilizing the context information (context-info) available in the 3D space. PointFusion [147] takes RGB and LIDAR images of a 2D detected object as input. The features extracted from the RGB channel using a CNN and from raw depth data using a PointNet are combined in a fusion net to regress 3D BBs.

Regression-based 3D object detectors directly reason about 3D BBs of the objects of interest. Deep SS [79] introduces the first fully conv 3D region proposal network (RPN), which converts the input depth volume into the 3D voxel grid and regresses $\mathbf{x}$ and $\mathbf{d}$. A joint object recognition network (ORN) scores the proposals taking their 3D volumes and corresponding RGB images. Both nets are trained based on the L_{1s} loss. The proposal-free, single stage 3D detection method of PIXOR [146] generates pixel-wise predictions, each of which corresponds to a 3D object estimate. As it operates on the Bird's Eye View (BEV) representation of LIDAR point cloud, it is amenable to real-time inference. VoxelNet [80], as an end-to-end trainable net, interfaces sparse LIDAR point clouds with RPN representing voxelized raw point cloud as sparse 4D tensors *via* a feature learning net. Using this representation, a Faster R-CNN-based RPN generates 3D detection, regressing the $\mathbf{x}$, $\mathbf{d}$, and θ_y parameters whose training is employed with L_{1s}. MV3D [82] processes the point cloud of LIDAR and acquires the BEV image. It firstly generates proposals regressing $\mathbf{x}$ and $\mathbf{d}$. Then, FusionNet [198] fuses the front view (FV) of LIDAR and RGB images of proposals to produce final detections regressing the 3D box corners $\{\mathbf{p}_{3Di}\}_{i=1}^{8}$. Liang et al. [143] is another end-to-end trainable method taking the advantages of both LIDAR and RGB information. It fuses RGB features onto the BEV feature map to regress the parameters $\mathbf{x}$, $\mathbf{d}$, and θ_y. Aggregate View Object Detection (AVOD) net [108] uses LIDAR point clouds and RGB images to generate features that are shared by two sub-nets. The first sub-net, an RPN, regresses the parameters $\mathbf{x}$ and $\mathbf{d}$ to produce 3D BB proposals, which are then fed to the second sub-net that refines the proposals further regressing $\{\mathbf{p}_{2Di}\}_{i=1}^{4}$, h_1, h_2, and θ_y.

The regression-based 6D object pose estimators are designed to estimate full 6D pose parameters. BB8 [40] firstly segments the object of interest in RGB image, and then feed the region into a CNN which is trained to predict the projections of 8 corners of the 3D BB. Full 6D pose is resolved applying a PnP algorithm the 2D-3D correspondences. The method additionally addresses the symmetry issue, narrowing down the range of rotation around the axis of symmetry of the training samples from 0 to the angle of symmetry γ. Tekin et al. [38] follow a similar approach to estimate objects' 6D pose, however, unlike BB8, they do not employ any segmentation and directly regress $\{\mathbf{p}_{3Di}^{proj}\}_{i=1}^{8}$ along with $\mathbf{p}_{2Dc}$. Oberweger et al. [163] present a method which is engineered to handle occlusion. The method extracts multiple patches from an image region centered on the object and predicts heatmaps over the 2D projections of 3D points. The heatmaps are aggregated and the 6D pose is estimated using a PnP algorithm. CDPN [167] disentangles the estimation of translation and rotation, separately predicting 3D coordinates for all image pixels and indirectly computing the rotation parameters from the predicted coordinates *via* PnP. Pix2Pose [168] addresses the challenges of the problem, i) occlusion: estimating the 3D coordinates per-pixel and generative adversarial training, ii) symmetry: introducing a novel L_1-based transformer loss, and iii) lack of precise 3D object models: using RGB images without textured models during training phase. Full 6D pose is obtained employing PnP & RANSAC over the predicted 3D coordinates. PVNet [160] estimates full 6D poses of the objects of interest under severe occlusion or truncation. To this end, it firstly regresses the vectors from pixels to the projections of the corners of the BBs, and then votes for the corner locations. Lastly, the 6D pose is computed *via* EPnP & Levenberg–Marquardt Algorithm (LMA). IHF [31], Sahin et al. [32], LCHF [4], and Doumanoglou et al. [35] directly regress the 6D pose parameters training Hough Forests (HF) with offset & pose entropy functions. IHF [31] and Sahin et al. [32] are engineered to work in depth images, while LCHF [4] and Doumanoglou et al. [35] take RGB-D as input. The 6D pose parameters
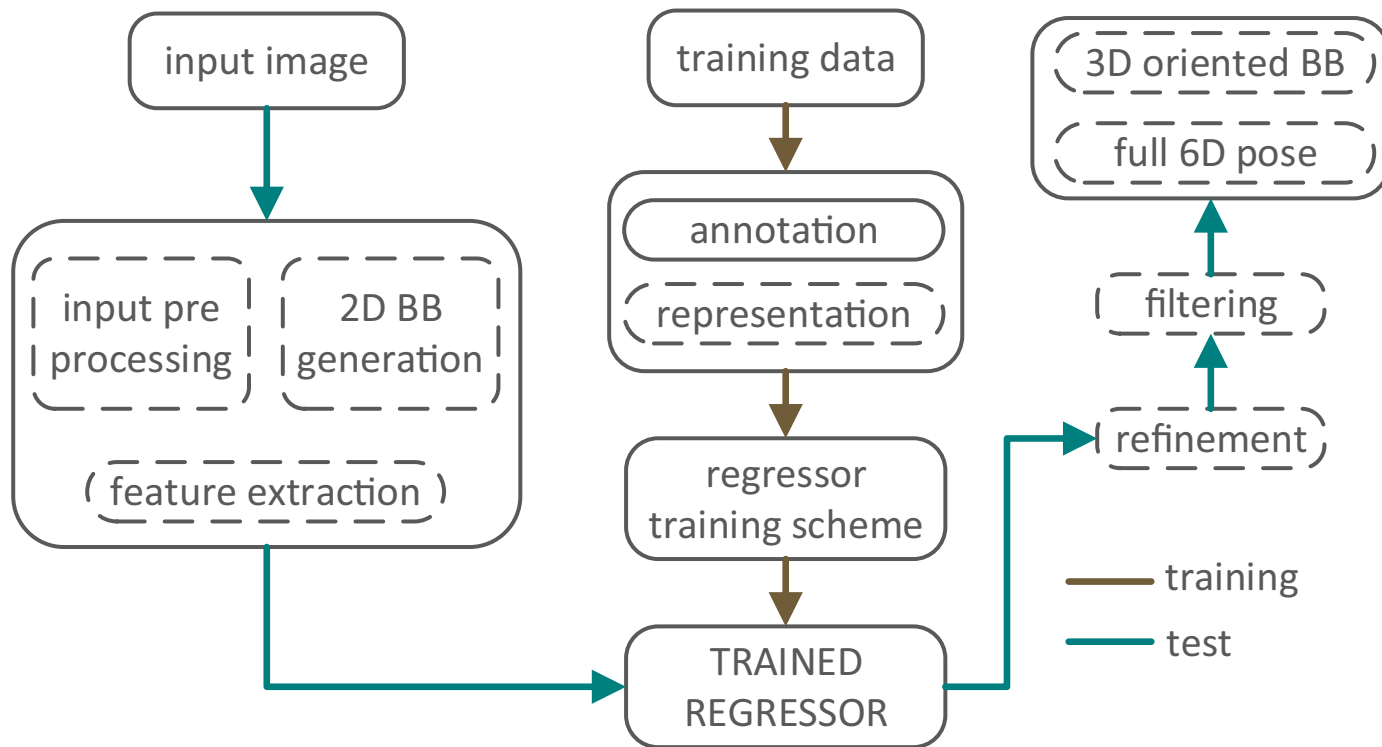
Fig. 2. Overall schematic representation of regression-based methods: blocks drawn with continuous lines are employed by all methods while dashed-line blocks are operated by clusters of specific methods (see the text and Table 3 for details).

Table 3
Regression-based methods (BEV: **B**ird's-**E**ye **V**iew, CNN: **C**onvolutional **N**eural **N**etwork, FV: **F**ront **V**iew, HF: **H**ough **F**orest, LMA: **L**evenberg–**M**arquardt **A**lgorithm, MLP: **M**ulti-**L**ayer **P**erceptron, Mo/St: **Mo**no/**St**ereo, NMS: **N**on-**M**aximum **S**uppression, ORN: **O**bject **R**ecognition **N**etwork, p-LIDAR: **p**seudo-**LIDAR**, EPnP: **E**fficient **P**erspective-**n**-**P**oint, R: **R**eal data, RPN: **R**egion **P**roposal **N**etwork, S: **S**ynthetic data, for symbols see Table 1).

Method	Input	Input pre-processing	Training data	Regression parameters	Regressor training	Trained regressor	Refinement step	Filtering	Level
2D-DRIVEN 3D									
Wang et al. [101]	RGB (Mo/St)	p-LIDAR & BEV	R	$\mathbf{x}, \mathbf{d}$	L_{1_s}	RPN	CNN scoring	NMS	Category
Refinement step	RGB (Mo/St)	p-LIDAR & BEV	R	$\{\mathbf{p}_{2D_i}\}_{i=1}^4, h_1, h_2, \theta_y$	L_{1_s}	CNN	✗	NMS	
Deng et al. [85]	RGB-D	✗	R	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	✗	NMS	Category
Lahoud et al. [83]	RGB-D	✗	R	$\mathbf{d}$	L_{1_s}	MLP	Context-info	✗	Category
PointFusion [147]	RGB-LIDAR	✗	R	$\{\mathbf{p}_{3D_i}\}_{i=1}$	L_{1_s}	CNN	✗	Scoring	Category
3D									
Deep SS [79]	Depth	3D grid	R	$\mathbf{x}, \mathbf{d}$	L_{1_s}	RPN (3D)	ORN	NMS	Category
Refinement step	RGB-D	3D grid	R	$\mathbf{x}, \mathbf{d}$	L_{1_s}	ORN	✗	Size pruning	
Li et al. [150]	LIDAR	Cylindrical	R	$\{\mathbf{p}_{3D_i}\}_{i=1}^8$	L_2	CNN	✗	NMS	Category
Bo Li [142]	LIDAR	3D grid	R	$\{\mathbf{p}_{3D_i}\}_{i=1}^8$	L_2	CNN	✗	NMS	Category
PIXOR [146]	LIDAR	BEV	R	x, y, d_w, d_l, θ_y	L_{1_s}	CNN	✗	NMS	Category
VoxelNet [80]	LIDAR	3D grid, BEV	R	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	RPN	✗	✗	Category
MV3D [82]	LIDAR	BEV	R	$\mathbf{x}, \mathbf{d}$	L_{1_s}	CNN	FusionNet	NMS	Category
Refinement step	RGB-LIDAR	BEV, FV	R	$\{\mathbf{p}_{3D_i}\}_{i=1}^8$	L_{1_s}	CNN	✗	NMS	
Liang et al. [143]	RGB-LIDAR	BEV	R	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	✗	NMS	Category
AVOD [108]	RGB-LIDAR	BEV	R	$\mathbf{x}, \mathbf{d}$	L_{1_s}	RPN	CNN scoring	NMS	Category
Refinement step	RGB-LIDAR	BEV	R	$\{\mathbf{p}_{2D_i}\}_{i=1}^4, h_1, h_2, \theta_y$	L_{1_s}	CNN	✗	NMS	
6D									
BB8 [40]	RGB	✗	R	$\{\mathbf{p}_{3D_i}^{proj}\}_{i=1}^8$	L_2	CNN	PnP	✗	Instance
Tekin et al. [38]	RGB	✗	R	$\{\mathbf{p}_{3D_i}^{proj}\}_{i=1}^8, \mathbf{p}_{2D_c}$	L_2	CNN	PnP	✗	Instance
Oberweger et al. [163]	RGB	✗	R & S	$\{\mathbf{p}_{3D_i}^{proj}\}_{i=1}^8$	L_2	CNN	PnP & RANSAC	✗	Instance
CDPN [167]	RGB	✗	R & S	$\mathbf{x}$	L_1	CNN	PnP & RANSAC	✗	Instance
Pix2Pose [168]	RGB	✗	R & S	$\mathbf{x}$	L_1	CNN	PnP & RANSAC	✗	Instance
Hu et al. [159]	RGB	✗	R & S	$\{\mathbf{p}_{3D_i}^{proj}\}_{i=1}^8$	L_1	CNN	EPnP	✗	Instance
PVNet [160]	RGB	✗	R & S	$\{\mathbf{p}_{3D_i}^{proj}\}_{i=1}^8$	L_{1_s}	CNN	EPnP & LMA	✗	Instance
IHF [31]	Depth	✗	S	$\mathbf{x}, \Theta$	Offset & pose entr	HF	Co-tra	✗	Instance
Sahin et al. [32]	Depth	✗	S	$\mathbf{x}, \Theta$	Offset & pose entr	HF	Co-tra	✗	Instance
LCHF [4]	RGB-D	✗	S	$\mathbf{x}, \Theta$	Offset & pose entr	HF	Co-tra	✗	Instance
Doumanoglou et al. [35]	RGB-D	✗	S	$\mathbf{x}, \Theta$	Offset & pose entr	HF	Joint reg	✗	Instance
DenseFusion [161]	RGB-D	✗	R	$\mathbf{x}, \Theta$	L_1	CNN	It ref	✗	Instance

are further refined conducting a co-training scheme [31,32,4] and employing joint registration (joint reg) [35].

The 2D-driven 3D methods, 3D methods, and the ones in [40,38,161] are trained using real data, while the training of the methods in [31,32,4,35] are conducted with synthetic renderings. [163,167,168,159,160] use both real and synthetic images to train. The 2D-driven 3D methods and the 3D BB detectors work at the level of categories, and the 6D methods [4,31,32,35,38,40,159–161,163,167,168] work at instance-level.

3.3. Classification & regression

Fig. 3 depicts the overall schematic representation of the classification & regression-based methods. Unlike the previous categories of methods, *i.e.*, classification-based and regression-based, this category performs the classification and regression tasks within a single architecture. The methods can firstly do the classification, the outcomes of which are cured in a regression-based refinement step [105,84,78,166] or *vice versa* [75], or can do the classification and regression in a single-shot process [87,145,101,106,100,148,103,102,30,37,162].

Table 4 expands on the classification & regression-based methods. Mousavian et al. [87] modify MS-CNN [123] so that the parameters $\mathbf{d}$ and θ_y are regressed in addition to the 2D BB of the object of interest. The regression of $\mathbf{d}$ is conducted by the L_2 loss, while the bin-based discrete-continuous loss is applied to firstly discretize θ_y into n overlapping bins, and then to regress the angle within each bin. The input of MonoPSR [106] is an RGB image, which is not subjected to any pre-processing. Once the 2D BB proposals for the the object of interest are generated using MS-CNN [123], MonoPSR hypothesizes 3D proposals, which are then fed into a *CNN scoring* refinement step. The scoring

net, in order to eliminate low confidence proposals and to produce final detection, regresses y and z. RGB-D-based method, FrustumPNet [100], detects 2D BBs in RGB images using FPN [119]. It does not pre-process the depth channel of the input RGB-D image, since it is capable of processing raw point clouds, as in PointNets [140,141]. Once the 2D BB proposals are gathered, they are lifted to the 3D space classifying and regressing $\mathbf{d}$ and θ_y, and regressing $\mathbf{x}$. The regression of $\mathbf{x}$ is provided training the net with L_{1s}, while $\mathbf{d}$ and θ_y are regressed using the bin-based version of L_{1s}, as employed in Faster R-CNN.

The method of Chen et al. [105] pre-processes the input RGB (St) images to obtain voxelized point clouds of the scene, and using this representation, detects the 3D BBs of the objects of interest in two stages: in the first stage, an SVM is trained to generate 3D BB proposals classifying
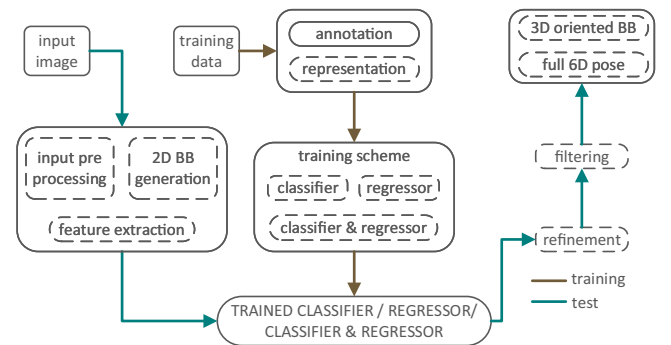

Fig. 3. Overall schematic representation of classification & regression-based methods: blocks drawn with continuous lines are employed by all methods while dashed-line blocks are operated by clusters of specific methods (see the text and Table 4 for details).

Table 4
Classification & Regression-based methods (BEV: **B**ird's-**E**ye **V**iew, CNN: **C**onvolutional **N**eural **N**etwork, CRF: **C**onditional **R**andom **F**ield, ICP: **I**terative **C**losest **P**oint, Mo/St: **Mo**no/**St**ereo, NMS: **N**on-**M**aximum **S**uppression, p-LIDAR: **p**seudo-**LIDAR**, PnP: **P**erspective-**n**-**P**oint, R: **R**eal data, RF: **R**andom **F**orest, S: **S**ynthetic data, SSD: **S**ingle-**S**hot **D**etector, s-SVM: **s**tructured-**S**upport **V**ector **M**achine, SVR: **S**upport **V**ector **R**egressor, for symbols see Table 1).

Method	Input	Input pre-processing	Training data	Classification parameters	Classifier training	Trained classifier	Regression parameters	Regressor training	Trained regressor	Refinement step	Filtering	Level
2D-DRIVEN 3D												
Mousavian et al. [87]	RGB	✗	R	θ_y	L_{ce}	CNN	$\mathbf{d}, \theta_y$	L_2	CNN	2D constraints	✗	Category
Xu et al. [145]	RGB	✗	R	θ_y	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	✗	✗	Category
Wang et al. [101]	RGB (Mo/St)	p-LIDAR	R	$\mathbf{d}, \theta_y$	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	FrustumPNet	✗	✗	Category
MonoPSR [106]	RGB	✗	R	θ_y	L_{ce}	CNN	$\mathbf{d}, \theta_y$	L_{1_s}	CNN	CNN scoring	✗	Category
Refinement step	RGB	✗	R	✗	✗	✗	y, z	L_{1_s}	CNN	✗	✗	
FrustumPNet [100]	RGB-D	✗	R	$\mathbf{d}, \theta_y$	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	PointNet	✗	✗	Category
3D												
Chen et al. [105]	RGB(St)	3D grid	R	$\mathbf{x}, \mathbf{d}, \theta_y$	IoU_{3D}	s-SVM	✗	✗	✗	CNN	NMS	Category
Refinement step	RGB (Mo)	✗	R	✗	✗	✗	x, y, d_w, d_h, θ_y	L_{1_s}	CNN	✗	NMS	
Chen et al. [84]	RGB	✗	R	$\mathbf{x}, \mathbf{d}, \theta_y$	IoU_{3D}	s-SVM	✗	✗	✗	CNN	NMS	Category
Refinement step	RGB	✗	R	✗	✗	✗	x, y, d_w, d_h, θ_y	L_{1_s}	CNN	✗	NMS	
DeepContext [78]	Depth	3D grid	R & S	$\mathbf{x}, \theta_y$	L_{ce}	CNN	✗	✗	✗	3DContextNet	✗	Category
Refinement step	Depth	3D grid	R & S	✗	✗	✗	$\mathbf{x}, \mathbf{d}$	L_{1_s}	CNN	✗	✗	
Lin et al. [75]	RGB-D	✗	R	✗	✗	✗	$\mathbf{x}$	L_{il}	SVR	CRF	NMS	Category
Refinement step	RGB-D	✗	R	$\mathbf{x}$	$L_{0/1}$	CRF	✗	✗	✗	✗	✗	
SECOND [148]	LIDAR	3D grid, BEV	R	θ_y	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	✗	✗	Category
PointPillars [103]	LIDAR	p-pillars	R	θ_y	L_{ce}	SSD	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	SSD	✗	NMS	Category
PointRCNN [102]	LIDAR	✗	R	x, z, θ_y	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	CNN, canonical ref	NMS	Category
Refinement step	LIDAR	✗	R	x, z, θ_y	L_{ce}	CNN	$\mathbf{x}, \mathbf{d}, \theta_y$	L_{1_s}	CNN	✗	NMS	
6D												
Brachmann et al. [30]	RGB	✗	R	$\mathbf{x}$	IG	RF	$\mathbf{x}$	L_1	RF	PnP & RANSAC & log-like	✗	Instance
SSD-6D [37]	RGB	✗	R & S	$\theta_{cvp}, \theta_{ip}$	L_{ce}	CNN	$\{\mathbf{p}_{2D,i}\}_{i=1}^4$	L_{1_s}	CNN	ICP	NMS	Instance
DPOD [166]	RGB	✗	R & S	$\mathbf{x}$	L_{ce}	CNN	✗	✗	✗	CNN	✗	Instance
Refinement step	RGB	✗	R & S	✗	✗	✗	$\mathbf{x}, \Theta$	L_1	CNN	✗	✗	
Li et al. [162]	RGB-D	✗	R & S	$\mathbf{x}, \Theta$	L_{ce}	CNN	$\mathbf{x}, \Theta$	L_2	CNN	Multi view	✗	Instance
Refinement step	RGB-D	✗	R & S	$\mathbf{x}, \Theta$	L_{ce}	CNN	$\mathbf{x}, \Theta$	L_2	CNN	ICP	✗	

$\mathbf{x}, \mathbf{d}$, and θ_y. In the second stage, the BB proposals are scored to produce final detections. Another work of Chen et al. [84] employs the same strategy as in [105], taking a monocular RGB image as input. The depth-based method, DeepContext [78], is formed of two CNNs. The first CNN is trained using L_{ce} to generate 3D BB proposals classifying $\mathbf{x}$ and θ_y. The proposals are then sent to the second CNN, which further regresses $\mathbf{x}$ and $\mathbf{d}$ to produce the final detections. RGB-D based method [75] first regresses $\mathbf{x}$ using Support Vector Regressor (SVR), which is further refined using a CRF whose training is based on the loss $L_{0/1}$. PointPillars [103] converts the input depth image of LIDAR into the pseudo image of point-pillars (p-pillars), and using this representation, outputs $\mathbf{x}, \mathbf{d}$, and θ_y to generate 3D BBs of the objects of interest. PointRCNN [102] detects the 3D BBs in two stages: it first directly processes raw depth images acquired from LIDAR sensors to produce box proposals. In the second stage, generated proposals are refined using another net which transforms the points of each proposal to canonical coordinates. The parameters and the loss functions of the first stage are used to produce final detection results.

Brachmann et al. [30] present a random forest-based (RF) architecture which lifts the 3D detection to 6D space employing RANSAC & log-likelihood algorithms. SSD-6D [37] simultaneously classifies θ_{cvp} and θ_{ip} and regresses the corners $\{\mathbf{p}_{2D,i}\}_{i=1}^4$ of 2D BBs. The training of

the net is based on L_{ce} and L_{1_s}. The output parameters of the net is lifted to the 6D space, employing an ICP-based refinement along with the utilization of the camera intrinsics.

The 2D-driven 3D methods and 3D methods are trained using real data. DeepContext [78] is trained on the partially synthetic training depth images which exhibit a variety of different local object appearances, and real data are used to fine tune the method. The 2D-driven 3D methods and the 3D BB detectors work at the level of categories, and the 6D methods [30,37,166,162] work at instance-level.

3.4. Template matching

The overall schematic representation of the template matching methods is given in Fig. 4.

3.4.1. Feature extraction phase

During an off-line feature extraction phase, 3D pose [180,34,41,181] or 6D pose [176,179,178,182,177,23,2,24,25] annotated templates involved in the training data are represented with robust feature descriptors. Features are manually-crafted utilizing the available shape, geometry, and appearance information [176,179,178,182,177,23,2,25],
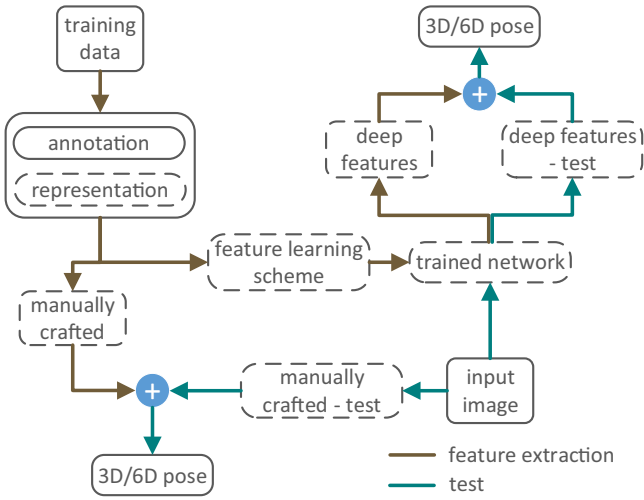
Fig. 4. Overall schematic representation of template matching methods: blocks drawn with continuous lines are employed by all methods while dashed-line blocks are operated by clusters of specific methods (see the text and Table 5 for details).

and the recent paradigm in the field is to deep learn those using neural net architectures [180,34,41,181].

3.4.2. Testing phase

A template-based method takes an image **I** as input on which it runs a sliding window during an on-line test phase. Each of the windows is represented with feature descriptors (either manually-crafted or deep features) and is compared with the templates stored in a memory. The matching is employed in the feature space, the distances between each window and the template set are computed. The pose parameters of a template are assigned to the window that has the closest distance with that template.

Table 5 details the template matching methods. Payet et al. [176] introduce Bag of Boundaries (BOB) representation, the histograms of the right contours belonging to the foreground object, and formulates the problem as matching the BOBs in the test image to the set of shape templates. The representation improves the robustness across clutter. Ulrich et al. [179] match test images to the set of templates using edge features. The method first estimates the discrete pose, which is further refined using the 2D match and the corresponding 3D camera pose in an LMA algorithm. Since the increase in the number of the templates in a database slows down the performance of template matching-based methods, Konishi et al. [178] present a method with a representation

of robust to change in appearance Perspectively Cumulated Orientation Feature (PCOF) and an efficient search algorithm of Hierarchical Pose Trees (HPT). 6D pose parameters initially estimated by edge matching-based Real-time Attitude and Position Determination-Linear Regressor (RAPID-LR) [182] are further refined by extracting HOG features on those edges and matching to the template database by RAPID-HOG distance. The method originally proposed for object tracking [177] can handle temporary tracking loss occurring when the object is massively occluded or is going out of the camera's viewpoint. It detects the object of interest matching the test image to the template database based on Temporally Consistent Local Color (TCLC) histograms. Liu et al. [23], estimating the 6D poses of objects merely taking depth image as input, extract depth edges. MTTM [180] firstly computes the segmentation mask in the test image and then employs template matching over RoIs within a CNN framework, where the pose parameters are encoded with quaternions. Linemod [2] and Hodan et al. [25] represent each pair of templates by surface normals and color gradients, while [24] trains SVMs to learn weights, which are then embedded into AdaBoost. Wohlhart et al. [34] define the triplet loss L_{tri} and the pairwise loss L_{pair} for training CNNs. L_{tri} enlarges the Euclidean distance between descriptors from two different objects and makes the Euclidean distance between descriptors from the same object representative of the similarity between their poses. L_{pair} makes the descriptors robust to noise and other distracting artifacts such as changing illumination. Balntas et al. [41] further improve L_{tri} and L_{pair} guiding the templates with pose during CNN training (L_{pose} loss). Zakharov et al. [181] introduce dynamic margin for the loss functions L_{tri} and L_{tri}, resulting faster training time and enabling the use of smaller-size descriptors with no loss of accuracy. The methods [34,41], and [181] are based on RGB-D, and the deep features are learnt based on synthetic and real data.

This family of the methods formulate the problem globally and represent the templates in the set and windows extracted from input images by feature descriptors holistically. Distortions along object borders arising from occlusion and clutter in the test processes mainly degrade the performance of these methods. Several imperfections of depth sensors, such as missing depth values, noisy measurements, *etc.* impair the surface representations at depth discontinuities, causing extra degradation in the methods' performance. Other drawback is matching features extracted during test to a set of templates, and hence, it cannot easily be generalized well to unseen ground truth annotations.

3.5. Point-pair feature matching

Fig. 5 represents the overall schematic layout of the point-pair feature matching methods. During an off-line phase, the global

Table 5
Template matching-based methods (BOB: **B**ag **O**f **B**oundaries, CNN: **C**onvolutional **N**eural **N**etwork, FDCM: **F**ast **D**irectional **C**hamfer **M**atching, HOG: **H**istogram of **O**riented **G**radients, HPT: **H**ierarchical **P**ose **T**ree, ICP: **I**terative **C**losest **P**oint, k-NN: **k**-**N**earest **N**eighbor, LMA: **L**evenberg–**M**arquardt **A**lgorithm, PCOF: **P**erspectively **C**umulated **O**rientation **F**eature, PnP: **P**erspective-**n**-**P**oint, PSO: **P**article **S**warm **O**ptimization, R: **R**eal data, RAPID-LR: **R**eal-time **A**ttitude and **P**os**I**tion **D**etermination-**L**inear **R**egressor, S: **S**ynthetic data, SVM: **S**upport **V**ector **M**achine, TCLC: **T**emporally **C**onsistent **L**ocal **C**olor histograms, for symbols see Table 1).

| Method	Input	Training data	Annotation	Manually crafted feature	Feat. learning scheme	Learnt features	Matching	Refinement step	Level
6D									
Payet et al. [176]	RGB	R	$\mathbf{x}, \Theta$	BOB	✗	✗	Convex opt	✗	Category
Ulrich et al. [179]	RGB	S	$\mathbf{x}, \Theta$	Edge	✗	✗	Dot pro	LMA	Instance
Konishi et al. [178]	RGB	S	z, Θ	PCOF	✗	✗	HPT	PnP	Instance
RAPID-LR [182]	RGB	S	$\mathbf{x}, \Theta$	Edge	✗	✗	RAPID-LR	RAPID-HOG	Instance
Refinement step	RGB	S	$\mathbf{x}, \Theta$	HOG	✗	✗	RAPID-HOG	✗	
Tjaden et al. [177]	RGB	R & S	$\mathbf{x}, \Theta$	TCLC	✗	✗	2D Euc	✗	Instance
Liu et al. [23]	Depth	S	$\mathbf{x}, \Theta$	Edge	✗	✗	FDCM	Multi-view	Instance
MTTM [180]	Depth	S	Θ	✗	L_{tri}, L_{pose}	CNN	NN	ICP	Instance
Linemod [2]	RGB-D	S	$\mathbf{x}, \Theta$	Normal, color grad	✗	✗	NN	ICP	Instance
Rios-Cabrera [24]	RGB-D	S	$\mathbf{x}, \Theta$	✗	SVM	Ada-Boost	NN	✗	Instance
Hodan et al. [25]	RGB-D	S	$\mathbf{x}, \Theta$	Normal, color grad	✗	✗	NN	PSO	Instance
Wohlhart et al. [34]	RGB-D	R & S	Θ	✗	L_{tri}, L_{pair}	CNN	kNN	✗	Instance
Balntas et al. [41]	RGB-D	R & S	Θ	✗	L_{tri}, L_{pose}	CNN	kNN	✗	Instance
Zakharov et al. [181]	RGB-D	R & S	Θ	✗	L_{tri_d}, L_{pair_d}	CNN	NN	✗	Instance
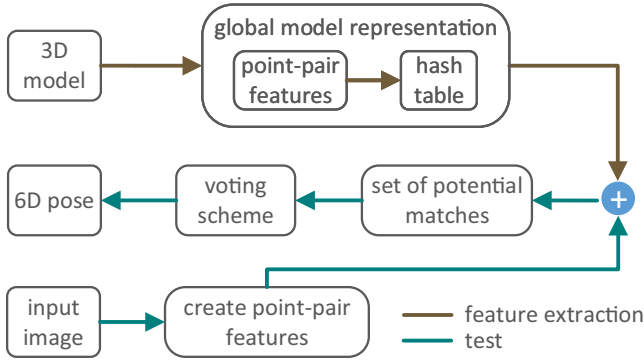
Fig. 5. Overall schematic representation of point-pair feature matching methods: (off-line phase) 3D model of an object of interest is globally represented with point-pair features (PPFs). (on-line phase) Point-pair features created from the input depth image are compared with the global model representation in order to estimate the 6D pose.

representation of the 3D model of an object of interest is formed by point-pair features (PPF) stored in a hash table. In an online phase, the point-pair features extracted in the test image are compared with the global model representation, from which the generated set of potential matches vote for the pose parameters.

We detail this category of methods initially focusing on the forefront publication from Drost et al. [17], and then presenting the approaches developed to improve PPF matching performance. Given two points $\mathbf{m}_1$ and $\mathbf{m}_2$ with normals $\mathbf{n}_1$ and $\mathbf{n}_2$, PPF is defined as follows [17]:

$$PPF(\mathbf{m}_1,,\mathbf{m}_2) = (\|\mathbf{dist}\|_2, \angle(\mathbf{n}_1,,\mathbf{dist}), \angle(\mathbf{n}_2,,\mathbf{dist}), \angle(\mathbf{n}_1,,\mathbf{n}_2)) \quad (5)$$

where $\mathbf{dist} = \mathbf{m}_2 - \mathbf{m}_1$, and $\angle(\mathbf{a},\mathbf{b}) \in [0;\pi]$ denotes the angle between two vectors $\mathbf{a}$ and $\mathbf{b}$. Hence, PPF describes the relative translation and rotation of two oriented points. The global description of the model M is created computing the feature vector PPF for all point pairs $\{(\mathbf{m}_r, \mathbf{m}_i)\} \in M$ and is represented as a hash table indexed by the feature vector $\mathbf{PPF}$ [16,17]. Similar feature vectors are grouped together so that they are located in the same bin of a hash table. Note that, global model description represents a mapping from the feature space to the model.

A set of point pair features are computed from the input test image $\mathbf{I}$: A reference point $\mathbf{s}_{ref}$ is firstly selected, and then all other points $\{\mathbf{s}_i\}$ available in $\mathbf{I}$ are paired with the reference point. Created point pairs $\{(\mathbf{s}_{ref}, \mathbf{s}_i)\}$ are compared with the ones stored in the global model representation. This comparison is employed in feature space, and a set of potential matches are resulted [14,16,17]. The rotation angle α between matched point pairs $(\mathbf{m}_r, \mathbf{m}_i)$ and $(\mathbf{s}_{ref}, \mathbf{s}_i)$ is calculated as follows:

$$\mathbf{s}_i = \mathbf{T}_s^{-1} \mathbf{R}_x(\alpha) \mathbf{T}_m \mathbf{m}_i \quad (6)$$

where $\mathbf{T}_s$ and $\mathbf{T}_m$ are the transformation matrices that translate $\mathbf{s}_{ref}$ and $\mathbf{m}_r$ into the origin and rotates their normals $\mathbf{n}_r^s$ and $\mathbf{n}_r^m$ onto the $\mathbf{x}$-axis [17]. Once the rotation angle α is calculated, the local coordinate $(\mathbf{m}_r, \alpha)$ is voted [14,16,17,21,22].

Initial attempts on improving the performance of point-pair matching methods have been on deriving novel features [15,18,19,20,21,136]. [15] augments the descriptors with the visibility context such as dimension, convexity, concativity to discard false matches in range data. [18], only relying on several scanned views of a target object, exploits color information for PPFs. Boundary points with directions and boundary line segments are also utilized [19]. Point-pair descriptors from both intensity and range are used along with geometric edge extractor to optimize surface and silhouette overlap of the scene and the model [20].

PPF matching methods show underperformance due to similar-looking distractors, occlusion, clutter, large planar surfaces, and sensor noise [201]. Their performance is further elevated by novel sampling and voting strategies [22,135,43]. The weighted voting function learnt in the max-margin learning framework [22] selects and ranks discriminative features by optimizing a discriminative cost function. [43] exploits object size for sampling to reduce the run-time and handles the sensor noise when voting using the neighboring bins of the lookup table. The run-time performance is also addressed in [137] paralleling the voting process. Recently presented deep architectures learn local descriptors from raw point clouds, which are encoded with PPFs [203] or additionally exploiting normal representations [204].

4. Datasets and metrics

In this section, we present the datasets most widely used to test the 3D BB detectors and full 6D pose estimators reviewed in this paper and detail the metrics utilized to measure the methods' performance.

4.1. Datasets

KITTI [110], SUN RGB-D [112], NYU-Depth v2 [111], and PASCAL3D+ [194] are the datasets on which the 3D BB detectors are commonly tested (see Table 6). KITTI has 14,999 images, 7481 of which are used to train the detectors, and the remaining is utilized to test. It has 3 discriminative categories, *car, pedestrian*, and *cyclist*, with a total number of 80,256 labeled objects. SUN RGB-D [112] has 10,335 RGB-D images in total, 3784 and 1159 of which are captured by Kinect v2 and IntelRealSense, respectively, while the remaining are collected from the datasets of NYU-Depth v2 [111], B3DO [185], and SUN3D [186]. For the 10,335 RGB-D images, there are 64,595 3D BB annotations with accurate orientations for about 800 object categories (2nd row of Table 6). There are around 5K training and 5K test images in the dataset for 37 object categories.[1] In the 2017 SUN RGB-D 3D object detection challenge,[2] existing 10,335 images along with their 64,595 3D BB annotations are used as training data, and newly acquired 2860 images, which have 3072 annotated object BBs, are utilized to test the methods' performance on 19 object classes (3rd row of Table 6). The NYU-Depth v2 dataset [111] involves 1449 images, 795 of which are of training, and the rest is for the test. Despite the fact that it has 894 object categories, there are mainly 19 object categories on which the methods are evaluated (*bathtub, bed, bookshelf, box, chair, counter, desk, door, dresser, garbage bin, lamp, monitor, nightstand, pillow, sink, sofa, table, tv, toilet*). 12 object categories of PASCAL VOC 2012 [7] (*aeroplane, bicycle, boat, bottle, bus, car, chair, dining table, motorbike, sofa, train,* and *tvmonitor*) are augmented with 3D annotations in PASCAL3D+ [194]. For each category, more images are added from ImageNet [6], resulting in a total number of 30,899 images with annotated objects. Apart from the datasets detailed in Table 6, we briefly mention other related 3D object detection datasets [188,189,190,191,192,193]. The RGB-D object dataset [188] involves 51 object classes, including 300 geometrically different object instances with clean background. The dataset in [189] provides indoor images with 3D annotations for several IKEA objects. NYC3DCars [190] and EPFL Cars [191] are presented for the *car* category. The images of the former are captured in the street scenes of New York City, and the latter one includes 2299 images of 20 car instances, which are taken at different azimuth angles but similar elevation and close distances. Table-Top Pose dataset [192] is formed of 480 images of 3 categories, *mouse, mug*, and *stapler*, and the one in [193] annotates the subsets of 4 categories of ImageNet, *bed, chair, sofa*, and *table*, with 3D BBs.

LINEMOD [2], MULT-I [4], OCC [28], BIN-P [35], and T-LESS [42] are the datasets most frequently used to test the performances of full 6D pose estimators. In a recently proposed benchmark for 6D object pose estimation [5], these datasets are refined and are presented in a unified format along with three new datasets (Rutgers Amazon Picking

---

[1] This statistics is obtained from [187].
[2] http://rgbd.cs.princeton.edu/challenge.html.

Table 6
Datasets utilized to test the reviewed 3D BB detectors (I/O: **I**ndoor/**O**utdoor, Mo/St: **Mo**no/**St**ereo).

Dataset	Modality	# Images	# Training	# Test	# Obj. classes	# Annotated obj.	I/O	Level
KITTI [110]	RGB(Mo/St), LIDAR	14,999	7481	7518	3	80,256	O	Category
SUN RGB-D v1 [112]	RGB-D	10,335	5285	5050	800	64,595	I	Category
SUN RGB-D v2 [112]	RGB-D	13,195	10,335	2860	19	67,667	I	Category
NYU-Depth v2 [111]	RGB-D	1449	795	654	894	35,064	I	Category
PASCAL3D + [194]	RGB	30,899	✗	✗	12	30,899	I & O	Category

Challenge (RU-APC) [184], TU Dresden Light (TUD-L), and Toyota Light (TYO-L)). The table which details the parameters of these datasets (Table 1, pp. 25, in [5]) is reformatted in our paper in Table 7 to present the datasets' challenges:

Viewpoint (VP) & Clutter (C). RU-APC [184] dataset involves the test scenes in which objects of interest are located at *varying viewpoints* and *cluttered backgrounds*.

VP & C & Texture-less (TL). Test scenes in the LINEMOD [2] dataset involve *texture-less* objects at varying viewpoints with cluttered backgrounds. There are 15 objects, for each of which more than 1100 real images are recorded. The sequences provide views from 0–360 degree around the object, 0–90 degree tilt rotation, ∓45 degree in-plane rotation, and 650 mm–1150 mm object distance.

VP & C & TL & Occlusion (O) & Multiple Instance (MI). Occlusion is one of the main challenges that makes the datasets more difficult for the task of object detection and 6D pose estimation. In addition to close and far range 2D and 3D clutter, testing sequences of the Multiple-Instance (MULT-I) dataset [4] contain *foreground occlusions* and *multiple object instances*. In total, there are approximately 2000 real images of 6 different objects, which are located at the range of 600 mm–1200 mm. The testing images are sampled to produce sequences that are uniformly distributed in the pose space by $[0^\circ, -360^\circ]$, $[-80^\circ, -80^\circ]$, and $[-70^\circ, -70^\circ]$ in the yaw, roll, and pitch angles, respectively.

VP & C & TL & Severe Occlusion (SO). Occlusion, clutter, texture-less objects, and change in viewpoint are the most well-known challenges that could successfully be dealt with the state-of-the-art 6D object detectors. However, *heavy existence* of these challenges severely degrades the performance of 6D object detectors. Occlusion (OCC) dataset [28] is one of the most difficult datasets in which one can observe up to 70 — 80% occluded objects. OCC includes the extended ground truth annotations of LINEMOD: in each test scene of the LINEMOD [2] dataset, various objects are present, but only ground truth poses for one object are given. Brachmann et al. [28] form OCC considering the images of one scene (benchvise) and annotating the poses of 8 additional objects.

VP & SC & SO & MI & Bin Picking (BP). In *bin-picking* scenarios, multiple instances of the objects of interest are arbitrarily stocked in a bin, and hence, the objects are inherently subjected to severe occlusion and severe clutter. Bin-Picking (BIN-P) dataset [35] is created to reflect such challenges found in industrial settings. It includes 177 test images of 2 textured objects under varying viewpoints.

VP & C & TL & O & MI & Similar-Looking Distractors (SLD). *Similar-looking distractor(s)* along with similar-looking object classes involved in the datasets strongly confuse recognition systems causing a lack of discriminative selection of shape features. Unlike the above-mentioned datasets and their corresponding challenges, the T-LESS [42] dataset particularly focuses on this problem. The RGB-D images of the objects located on a table are captured at different viewpoints covering 360 degrees rotation, and various object arrangements generate occlusion. Out-of-training objects, similar-looking distractors (planar surfaces), and similar-looking objects cause 6 DoF methods to produce many false positives, particularly affecting the depth modality features. T-LESS has 30 texture-less industry-relevant objects, and 20 different test scenes, each of which consists of 504 test images.

VP & Light (LI). TUD-L is a dedicated dataset for measuring the performances of the 6D pose estimators across different ambient and *lighting* conditions. The objects of interest are subjected to 8 different lighting conditions in the test images, in the first frame of which object pose is manually aligned to the scene using 3D object model. The initial pose is then propagated through the sequence using ICP.

VP & C & LI. As in the TUD-L dataset, different lighting condition is also featured in TYO-L dataset, where the target objects are located in a cluttered background. Its 21 objects are captured on a table-top setup under 5 different illumination.

We also briefly mention about new datasets which are utilized on measuring the performance of full 6D pose estimators. YCB-Video dataset [39] is formed of 92 video sequences, 80 of which are used for training, while the remaining sequences are of the testing phase. The total number of real images are about 133K, and it additionally contains around 80K synthetically rendered images to be utilized for training. The test images, where 21 objects taken from the YCB dataset [195,196] exist, are subjected to significant image noise, varying lighting conditions, and occlusion in cluttered background. JHUScene-50 [197] dataset contains 5 different indoor environments including office workspaces, robot manipulation platforms, and large containers. Each environment contains 10 scenes, each of which is composed of 100 test frames with densely cluttered multiple object instances. There are 10 hand tool objects, 5000 test images, and 22,520 labeled poses in total. Sample datasets for 6D object pose recovery are shown in Fig. 6.

Table 7
Datasets used to test the reviewed 6D pose estimators (C: **C**lutter, MI: **M**ultiple **I**nstance, O: **O**cclusion, SC: **S**evere **C**lutter, SO: **S**evere **O**cclusion, VP: **V**iew**P**oint, SLD: **S**imilar-**L**ooking **D**istractors, BP: **B**in **P**icking, LI: **LI**ght).

Dataset	Challenge	# Objects	Modality	# Total frame	Level
RU-APC [184]	VP & C	14	RGB-D	5964	Instance
LINEMOD [2]	VP & C & TL	15	RGB-D	18,273	Instance
MULT-I [4]	VP & C & TL & O & MI	6	RGB-D	2067	Instance
OCC [28]	VP & C & TL & SO	8	RGB-D	8916	Instance
BIN-P [35]	VP & SC & SO & MI & BP	2	RGB-D	177	Instance
T-LESS [42]	VP & C & TL & O & MI & SLD	30	RGB-D	10,080	Instance
TUD-L	VP & LI	3	RGB-D	23,914	Instance
TYO-L	VP & C & LI	21	RGB-D	1680	Instance

4.2. Metrics: correctness of an estimation

Several evaluation metrics have been proposed to determine the correctness of 3D BB estimations and full 6D pose hypotheses.

Fig. 6. Sample datasets for 6D object pose recovery. Row-wise, the 1st benchmark concerns texture-less objects at varying viewpoint with cluttered background, the 2nd is interested in multi-instance, the 3rd has scenes with severely occluded objects, the 4th reflects the challenges found in bin-picking scenarios.

4.2.1. Metrics for 3D bounding boxes

We firstly present the metrics for 3D BB detectors.

4.2.1.1. Intersection over Union (IoU) [7].
This metric is originally presented to evaluate the performance of the methods working in 2D space [7]. Given the estimated and ground truth BBs B and $\overline{B}$ and assuming that they are aligned with image axes, it determines the area of intersection $B \cap \overline{B}$, and the area of union $B \cup \overline{B}$, and then comparing these two, outputs the overlapping ratio $\omega_{IoU_{2D}}$:

$$\omega_{IoU_{2D}} = \frac{B \cap \overline{B}}{B \cup \overline{B}} \tag{7}$$

According to Eq. (7), a predicted box is considered to be correct (true positive) if the overlapping ratio ω_{IoU2D} is more than the threshold $\tau_{IoU2D} = 0.5$ [7]. This metric is further extended to work with 3D volumes calculating overlapping ratio ω_{IoU3D} over 3D BBs [89]. The extended version assumes that 3D BBs are aligned with gravity direction, but makes no assumption on the other two axes.

4.2.1.2. Average precision (AP) [7].
It depicts the shape of the Precision/Recall (PR) curve. Dividing the recall range $r = [0\ 1]$ into a set of 11 equal levels, it finds the mean precision p at this set [7]:

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,...,1\}} p_{interp}(r) \tag{8}$$

The precision p at each recall level r is interpolated by taking the maximum precision measured for a method for which the corresponding recall exceeds r:

$$p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{9}$$

where $p(\tilde{r})$ is measured precision at recall $\tilde{r}$.

4.2.1.3. Average orientation similarity (AOS) [110].
This metric firstly measures the object detection performance of any detector in 2D image plane using the AP metric in [7], and then, evaluates the joint detection and 3D orientation estimation performance as follows [110]:

$$AOS = \frac{1}{11} \sum_{r \in \{0,0.1,...,1\}} \max_{\tilde{r}:\tilde{r} \geq r} sim(\tilde{r}), \tag{10}$$

where, the orientation similarity at recall r, $sim(r)$, is a normalized ($[0...1]$) version of the cosine similarity given below:

$$sim(r) = \frac{1}{|\mathcal{D}(r)|} \sum_{i \in \mathcal{D}(r)} \frac{1 + \cos\Delta_\theta^{(i)}}{2} \delta_i. \tag{11}$$

In Eq. (11), $\mathcal{D}(r)$ depicts the set of all object detections at recall level r, $\Delta_\theta^{(i)}$ is the difference in angle between estimated and ground truth orientation of detection i. δ_i is set to 1 if detection i is assigned to a ground truth BB (overlaps by at least 50%), and $\delta_i = 0$ if it has not been assigned. This penalizes multiple detections which explain a single object.

4.2.1.4. Average viewpoint precision (AVP) [194].
As in [110], this metric measures the performance of any detector jointly considering 2D object detection and 3D orientation. An estimation is accepted as correct if and

only if the BB overlap is larger than 50% and the viewpoint is correct (*i.e.*, the two viewpoint labels are the same in discrete viewpoint space or the distance between the two viewpoints is smaller than some threshold in continuous viewpoint space). Then, a Viewpoint Precision-Recall (VPR) curve, the area under of which is equal to average viewpoint precision, is drawn.

4.2.2. Metrics for full 6D pose

Once we detail the metrics for 3D BB detectors, we next present the metrics for the methods of full 6D pose estimators.

4.2.2.1. Translational and angular error. As being independent from the models of objects, this metric measures the correctness of a hypothesis according to the followings [17]: i) $\mathcal{L}_2$ norm between the ground truth and estimated translations $\bar{\mathbf{x}}$ and $\mathbf{x}$, ii) the angle computed from the axis-angle representation of ground truth and estimated rotation matrices $\bar{\mathbf{R}}$ and $\mathbf{R}$:

$$\omega_{\text{TE}} = \|(\bar{\mathbf{x}} - \mathbf{x})\|_2, \tag{12}$$

$$\omega_{\text{RE}} = \arccos\left(Tr\left(\mathbf{R}\bar{\mathbf{R}}^{-1} - 1\right)/2\right). \tag{13}$$

According to Eqs. (12) and (13), a hypothesis is accepted as correct if the scores ω_{TE} and ω_{RE} are below predefined thresholds τ_{TE} and τ_{RE}.

4.2.2.2. 2D projection. This metric takes the 3D model of an object of interest as input and projects the model's vertices $\mathbf{v}$ onto the image plane at both ground truth and estimated poses. The distance of the projections of corresponding vertices is determined as follows:

$$\omega_{\text{2Dproj}} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{v} \in \mathcal{V}} \|\mathbf{K}\bar{\mathbf{R}}\mathbf{v} - \mathbf{K}\mathbf{R}\mathbf{v}\|_2 \tag{14}$$

where $\mathcal{V}$ is the set of all object model vertices, $\mathbf{K}$ is the camera matrix. Homogenous coordinates are normalized before calculating the $\mathcal{L}_2$ norm. An estimated pose is accepted as correct if the average reprojection error ω_{2Dproj} is below 5px. As this error is basically computed using the 3D model of an object of interest, it can also be calculated using its point cloud.

4.2.2.3. Average Distance (AD). This is one of the most widely used metrics in the literature [2]. Given the ground truth $(\bar{\mathbf{x}}, \bar{\mathbf{\Theta}})$ and estimated $(\mathbf{x}, \mathbf{\Theta})$ poses of an object of interest O, this metric outputs ω_{AD}, the score of the AD between $(\bar{\mathbf{x}}, , \bar{\mathbf{\Theta}})$ and $(\mathbf{x}, \mathbf{\Theta})$. It is calculated over all points $\mathbf{s}$ of the 3D model M of the object of interest:

$$\omega_{\text{AD}} = \underset{\mathbf{s} \in M}{\text{avg}} \|(\bar{\mathbf{R}}\mathbf{s} + \bar{\mathbf{T}}) - (\mathbf{R}\mathbf{s} + \mathbf{T})\| \tag{15}$$

where $\bar{\mathbf{R}}$ and $\bar{\mathbf{T}}$ depict rotation and translation matrices of the ground truth pose $(\bar{\mathbf{x}}, , \bar{\mathbf{\Theta}})$, while $\mathbf{R}$ and $\mathbf{T}$ represent rotation and translation matrices of the estimated pose $(\mathbf{x}, \mathbf{\Theta})$. Hypotheses ensuring the following inequality are considered as correct:

$$\omega_{\text{AD}} \le z_\omega \Phi \tag{16}$$

where Φ is the diameter of the 3D model M, and z_ω is a constant that determines the coarseness of a hypothesis which is assigned as correct. Note that, Eq. (15) is valid for objects whose models are not ambiguous or do not have any subset of views under which they appear to be ambiguous. In case the model M of an object of interest has indistinguishable views, Eq. (15) transforms into the following form:

$$\omega_{\text{AD}} = \underset{\mathbf{s_1} \in M}{\text{avg}} \underset{\mathbf{s_2} \in M}{\min} \|(\bar{\mathbf{R}}\mathbf{s_1} + \bar{\mathbf{T}}) - (\mathbf{R}\mathbf{s_2} + \mathbf{T})\| \tag{17}$$

where ω_{AD} is calculated as the AD to the closest model point. This

function employs many-to-one point matching and significantly promotes symmetric and occluded objects, generating lower ω_{AD} score.

4.2.2.4. Visible Surface Discrepancy (VSD). This metric has recently been proposed to eliminate ambiguities arising from object symmetries and occlusions [3]. The model M of an object of interest O is rendered at both ground truth $(\bar{\mathbf{x}}, \bar{\mathbf{\Theta}})$ and estimated $(\mathbf{x}, \mathbf{\Theta})$ poses, and the depth maps $\bar{\mathbf{D}}$ and $\mathbf{D}$ of the renderings are intersected with the test depth image itself $\mathbf{I_D}$ in order to generate the visibility masks $\bar{\mathbf{V}}$ and $\mathbf{V}$. By comparing the generated masks, ω_{VSD}, the score that determines whether an estimation is correct, according to a pre-defined threshold τ, is computed as follows:

$$\omega_{\text{VSD}} = \underset{q \in \bar{\mathbf{V}} \cup \mathbf{V}}{\text{avg}} \begin{cases} 0 \text{ if } q \in \bar{\mathbf{V}} \cap \mathbf{V} \wedge \mid \bar{\mathbf{D}}(q) - \mathbf{D}(q) \mid < \tau \\ 1 \qquad\qquad\quad \text{otherwise} \end{cases} \tag{18}$$

where q depicts pixel. As formulized, the score ω_{VSD} is calculated only over the visible part of the model surface, and thus the indistinguishable poses are treated as equivalent.

4.2.2.5. Sym pose distance. VSD [3] is presented as an ambiguity-invariant pose error function. The acceptance of a pose hypothesis as correct depends on the plausibility of the estimated pose given the available data. However, numerous applications, particularly in the context of active vision, rely on a precise pose estimation and cannot be satisfied by plausible hypotheses, and hence VSD is considered problematic for evaluation. When objects of interest are highly occluded, the number of plausible hypotheses can be infinitely large. *Sym* pose distance deals with the ambiguities arising from the symmetry issue under a theoretically backed framework. Within this framework, this metric outputs the score ω_{Sym} between poses $\mathcal{P}_1$ and $\mathcal{P}_2$:

$$\omega_{Sym} = \underset{\mathbf{G_1}, \mathbf{G_2} \in G}{\min} \sqrt{\frac{1}{A} \int_A \|\mathbf{W_2} \cdot \mathbf{G_2}(\mathbf{x}) - \mathbf{W_1} \cdot \mathbf{G_1}(\mathbf{x})\|^2 da}, \tag{19}$$

where the pose $\mathcal{P}_1$ is identified to the equivalency class $\{\mathbf{W_1} \circ \mathbf{G_1}(\mathbf{x}) | \mathbf{G_1} \in G\}$, and the pose $\mathcal{P}_2$ is identified to the equivalency class $\{\mathbf{W_2} \circ \mathbf{G_2}(\mathbf{x}) | \mathbf{G_2} \in G\}$. A is the surface area of the object of interest, $\mathbf{W_1}$ and $\mathbf{W_2}$ are rigid transformations, and $\mathbf{G_1}$ and $\mathbf{G_2}$ are the symmetry groups in $G \in SE(3)$.

4.2.2.6. Implementation details. In this study, we employ a twofold evaluation strategy for the 6D detectors using both AD and VSD metrics: i) Recall. The hypotheses on the test images of every object are ranked, and the hypothesis with the highest weight is selected as the estimated 6D pose. Recall value is calculated comparing the number of correctly estimated poses and the number of the test images of the interested object. ii) F1 scores. Unlike recall, all hypotheses are taken into account, and F1 score, the harmonic mean of precision and recall values, is presented.

5. Multi-modal analyses

We analyze 17 baselines on the datasets with respect to both challenges and the architectures. Two of the baselines [2,4] are our own implementations. The color gradients and surface normal features, presented in [2], are computed using the built-in functions and classes provided by OpenCV. The features in LCHF [4] are the part-based version of the features introduced in [2]. Hence, we inherit the classes given by OpenCV in order to generate part-based features used in LCHF. We train each method for the objects of interest by ourselves, and using the learnt detectors, we test those on all datasets. Note that, the methods use only foreground samples during training/template generation. In this section, LINEMOD refers to the dataset, whilst Linemod is used to indicate the baseline itself.

5.1. Analyses based on Average Distance

Utilizing the AD metric, we compare the chosen baselines along with the challenges, i) regarding the recall values that each baseline generates on every dataset, ii) regarding the F1 scores. The coefficient z_ω is 0.10, and in case we use different thresholds, we will specifically indicate in the related parts.

5.1.1. Recall-only discussions

Recall-only discussions are based on the numbers provided in Table 8 and Fig. 7.

5.1.1.1. Clutter, viewpoint, texture-less objects. Highest recall values are obtained on the LINEMOD dataset (see Fig. 7), meaning that the state-of-the-art methods for 6D object pose estimation can successfully handle the challenges, clutter, varying viewpoint, and texture-less objects. LCHF, detecting more than half of the objects with over 80% accuracy, worst performs on box and glue (see Table 8(a)), since these objects have planar surfaces, which confuses the features extracted in depth channel (example images are given in Fig. 8(a)).

5.1.1.2. Occlusion. In addition to the challenges involved in LINEMOD, occlusion is introduced in MULT-I. Linemod's performance decreases, since occlusion affects holistic feature representations in color and depth channels. LCHF performs better on this dataset than Linemod. Since LCHF is trained using the parts coming from positive training images, it can easily handle occlusion, using the information acquired from occlusion-free parts of the target objects. However, LCHF degrades on camera. In comparison with the other objects in the dataset, camera has relatively smaller dimensions. In most of the test images, there are non-negligible amount of missing depth pixels (Fig. 8(b)) along the borders of this object, and thus confusing the features extracted in depth channel. In such cases, LCHF is liable to detect similar-looking out of training objects and generate many false positives (see Fig. 8(c)). The hypotheses produced by LCHF for joystick are all considered as false positive (Fig. 8(d)). When we re-evaluate the recall that LCHF produces on the joystick object setting z_ω to the value of 0.15, we observe 89% accuracy.

5.1.1.3. Severe occlusion. OCC involves challenging test images where the objects of interest are cluttered and severely occluded. The best performance on this dataset is caught by PoseCNN [39], and there is still room for improvement in order to fully handle this challenge. Despite the fact that the distinctive feature of this benchmark is the existence of severe occlusion, there are occlusion-free target objects in several test images. In case the test images of a target object include unoccluded and/or naively occluded samples (with the occlusion ratio up to 40% — 50% of the object dimensions) in addition to severely occluded samples, methods produce relatively higher recall values (*e.g.*, can, driller, duck, holepuncher, Table 8(c)). On the other hand, when the target object has additionally other challenges such as planar surfaces, methods' performance (LCHF and Linemod) decreases (*e.g.*, box, Fig. 8(e)).

Table 8
Methods' performance are depicted object-wise based on recall values computed using the Average Distance (AD) evaluation protocol.

(a) LINEMOD

Method	Input	Ape	Bvise	Cam	Can	Cat	Driller	Duck	Box	Glue	Hpuncher	Iron	Lamp	Phone	Aver
Brach et al. [30]	RGB-D	98.1	99.0	99.7	99.7	99.1	100	96.2	99.7	99.0	98.0	99.9	99.5	99.6	99.0
Kehl et al. [36]	RGB-D	96.9	94.1	97.7	95.2	97.4	96.2	97.3	99.9	78.6	96.8	98.7	96.2	92.8	95.2
Brach et al. [28]	RGB-D	85.4	98.9	92.1	84.4	90.6	99.7	92.7	91.1	87.9	97.9	98.8	97.6	86.1	92.6
LCHF [4]	RGB-D	84.0	95.0	72.0	74.0	91.0	92.0	91.0	48.0	55.0	89.0	72.0	90.0	69.0	78.6
Cabrera et al. [24]	RGB-D	95.0	98.9	98.2	96.3	99.1	94.3	94.2	99.8	96.3	97.5	98.4	97.9	88.3	96.5
Linemod [2]	RGB-D	95.8	98.7	97.5	95.4	99.3	93.6	95.9	99.8	91.8	95.9	97.5	97.7	93.3	96.3
Hodan et al. [25]	RGB-D	93.9	99.8	95.5	95.9	98.2	94.1	94.3	100	98.0	88.0	97	88.8	89.4	94.9
Hashmod [44]	RGB-D	95.6	91.2	95.2	91.8	96.1	95.1	92.9	99.9	95.4	95.9	94.3	94.9	91.3	94.6
Hinters et al. [43]	RGB-D	98.5	99.8	99.3	98.7	99.9	93.4	98.2	98.8	75.4	98.1	98.3	96.0	98.6	96.4
Drost et al. [17]	Depth	86.5	70.7	78.6	80.2	85.4	87.3	46.0	97.0	57.2	77.4	84.9	93.3	80.7	78.9
SSD-6D [37]	RGB	65.0	80.0	78.0	86.0	70.0	73.0	66.0	100	100	49.0	78.0	73.0	79.0	76.7
BB8 [40]	RGB	40.4	91.8	55.7	64.1	62.6	74.4	44.3	57.8	41.2	67.2	84.7	76.5	54.0	62.7
Tekin et al. [38]	RGB	21.6	81.8	36.6	68.8	41.8	63.5	27.2	69.6	80.0	42.6	75.0	71.1	47.7	56.0
Brach et al. [30]	RGB	33.2	64.8	38.4	62.9	42.7	61.9	30.2	49.9	31.2	52.8	80.0	67.0	38.1	50.3

(b) MULT-I

Method	Input	Camera	Cup	Joystick	Juice	Milk	Shampoo	Aver
LCHF [4]	RGB-D	52.5	99.8	98.3	99.3	92.7	97.2	90.0
Linemod [2]	RGB-D	18.3	99.2	85.0	51.6	72.2	53.1	63.2

(c) OCC

Method	Input	Ape	Can	Cat	Driller	Duck	Box	Glue	Hpuncher	Aver
PoseCNN [39]	RGB-D	76.2	87.4	52.2	90.3	77.7	72.2	76.7	91.4	78.0
Michel et al. [33]	RGB-D	80.7	88.5	57.8	94.7	74.4	47.6	73.8	96.3	76.8
Krull et al. [29]	RGB-D	68.0	87.9	50.6	91.2	64.7	41.5	65.3	92.9	70.3
Brach et al. [28]	RGB-D	53.1	79.9	28.2	82.0	64.3	9.0	44.5	91.6	56.6
LCHF [4]	RGB-D	48.0	79.0	38.0	83.0	64.0	11.0	32.0	69.0	53.0
Hinters et al. [43]	RGB-D	81.4	94.7	55.2	86.0	79.7	65.5	52.1	95.5	76.3
Linemod [2]	RGB-D	21.0	31.0	14.0	37.0	42.0	21.0	5.0	35.0	25.8
PoseCNN [39]	RGB	9.6	45.2	0.93	41.4	19.6	22.0	38.5	22.1	25.0

(d) BIN-P

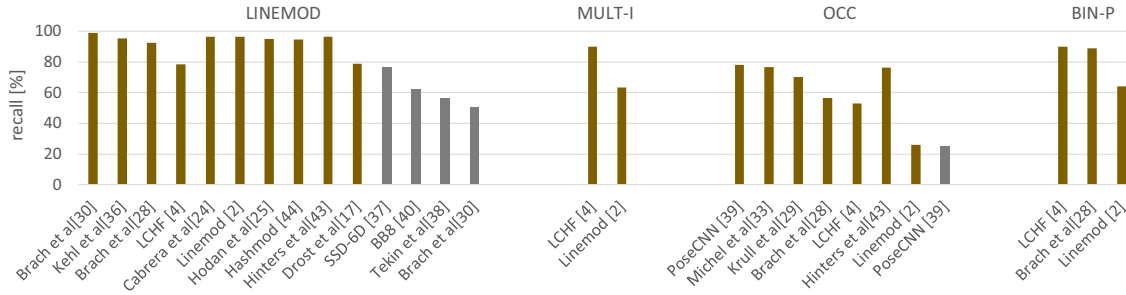| Method	Input	Cup	Juice	Aver
LCHF [4]	RGB-D	90.0	89.0	90.0
Brach et al. [28]	RGB-D	89.4	87.6	89.0
Linemod [2]	RGB-D	88.0	40.0	64.0

Fig. 7. Success of each baseline on every dataset is shown, recall values are computed using the Average Distance (AD) metric.

5.1.1.4. Severe clutter. In addition to the challenges discussed above, BIN-P inherently involves severe clutter, since it is designed for bin-picking scenarios, where objects are arbitrarily stacked in a pile. According to the recall values presented in Table 8(d), LCHF and Brachmann et al. [28] perform 25% better than Linemod. Despite having severely occluded target objects in this dataset, there are unoccluded/relatively less occluded objects at the top of the bin. Since our current analyses are based on the top hypothesis of each method, the produced success rates show that the methods can recognize the objects located on top of the bin with reasonable accuracy (Fig. 8(f)).

5.1.1.5. Similar-looking distractors. We test both Linemod and LCHF on the T-LESS dataset. Since most of the time the algorithms fail, we do not report quantitative analyses, instead we discuss our observations from the experiments. The dataset involves various object classes with strong shape and color similarities. When the background color is different than that of the objects of interest, color gradient features are successfully extracted. However, the scenes involve multiple instances, multiple objects similar in shape and color, and hence, the features queried exist in the scene at multiple locations. The features extracted in depth channel are also severely affected from the lack of discriminative

(a) planar surface

(b) missing depth pixels

(c) out of training samples

(d) false positive consideration

(e) severe occlusion

(f) detecting top objects

(g) similar-looking dist.
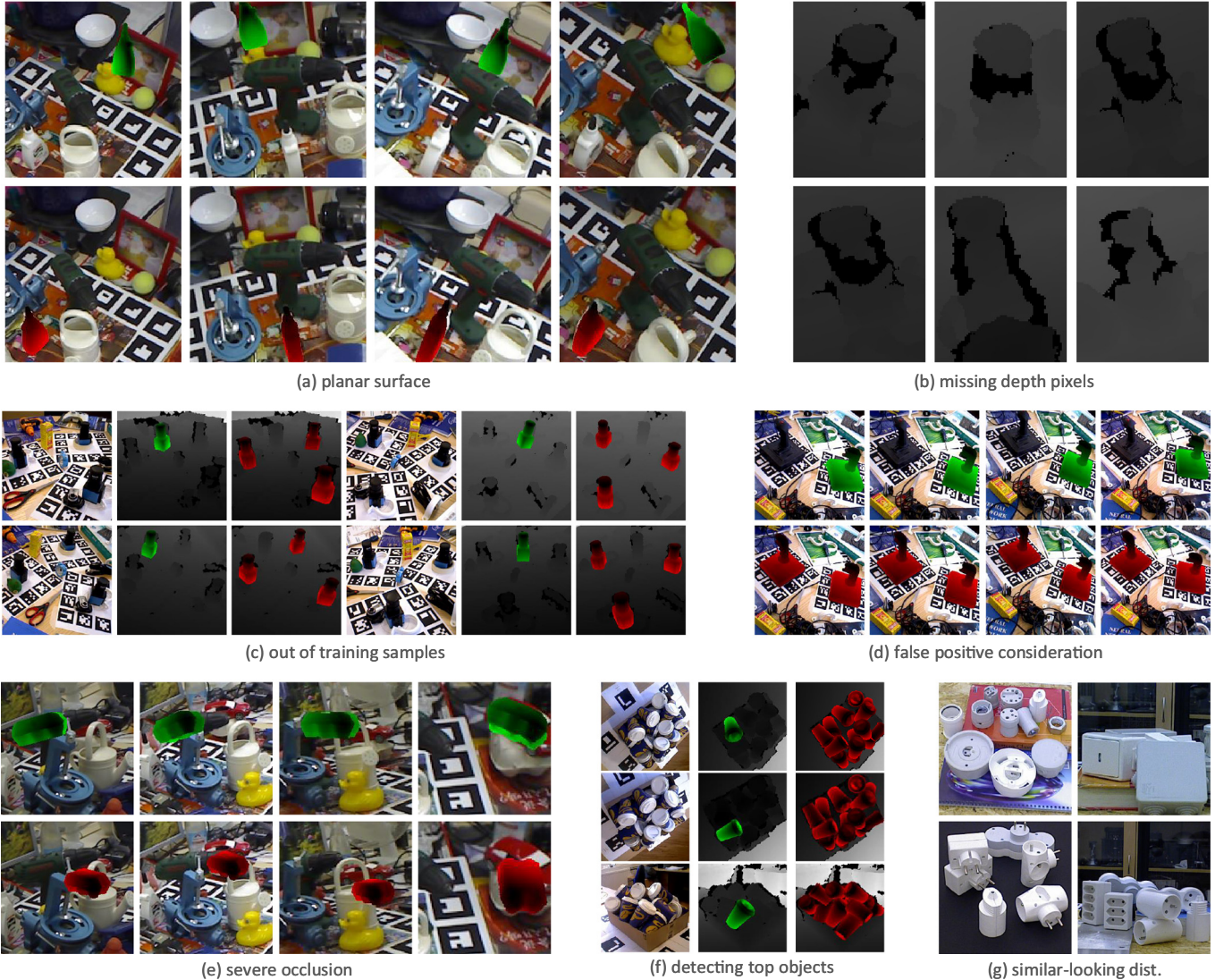
Fig. 8. Challenges encountered during test are exemplified (green renderings are hypotheses, and the red ones are ground truths). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 9
Methods' performance are depicted object-wise based on F1 scores computed using the Average Distance (AD) evaluation protocol.

(a) LINEMOD

Method	Input	Ape	Bvise	Cam	Can	Cat	Driller	Duck	Box	Glue	Hpuncher	Iron	Lamp	Phone	Aver
Kehl et al. [36]	RGB-D	0.98	0.95	0.93	0.83	0.98	0.97	0.98	1	0.74	0.98	0.91	0.98	0.85	0.93
LCHF [4]	RGB-D	0.86	0.96	0.72	0.71	0.89	0.91	0.91	0.74	0.68	0.88	0.74	0.92	0.73	0.82
Linemod [2]	RGB-D	0.53	0.85	0.64	0.51	0.66	0.69	0.58	0.86	0.44	0.52	0.68	0.68	0.56	0.63
SSD-6D [37]	RGB	0.76	0.97	0.92	0.93	0.89	0.97	0.80	0.94	0.76	0.72	0.98	0.93	0.92	0.88

(b) MULT-I

Method	Input	Camera	Cup	Joystick	Juice	Milk	Shampoo	Aver
Doum et al. [35]	RGB-D	0.93	0.74	0.92	0.90	0.82	0.51	0.8
Kehl et al. [36]	RGB-D	0.38	0.97	0.89	0.87	0.46	0.91	0.75
LCHF [4]	RGB-D	0.39	0.89	0.55	0.88	0.40	0.79	0.65
Drost et al. [17]	RGB-D	0.41	0.87	0.28	0.60	0.26	0.65	0.51
Linemod [2]	RGB-D	0.37	0.58	0.15	0.44	0.49	0.55	0.43
SSD-6D [37]	RGB	0.74	0.98	0.99	0.92	0.78	0.89	0.88

(c) OCC

Method	Input	Ape	Can	Cat	Driller	Duck	Box	Glue	Hpuncher	Aver
LCHF [4]	RGB-D	0.51	0.77	0.44	0.82	0.66	0.13	0.25	0.64	0.53
Linemod [2]	RGB-D	0.23	0.31	0.17	0.37	0.43	0.19	0.05	0.30	0.26
Brach et al. [30]	RGB	✗	✗	✗	✗	✗	✗	✗	✗	0.51
Tekin et al. [38]	RGB	✗	✗	✗	✗	✗	✗	✗	✗	0.48
SSD-6D [37]	RGB	✗	✗	✗	✗	✗	✗	✗	✗	0.38

(d) BIN-P

Method	Input	Cup	Juice	Aver
LCHF [4]	RGB-D	0.48	0.29	0.39
Doum et al. [35]	RGB-D	0.36	0.29	0.33
Linemod [2]	RGB-D	0.48	0.20	0.34

selection of shape information. When the objects of interest have planar surfaces, the detectors cannot easily discriminate foreground and background in depth channel, since these objects in the dataset are relatively smaller in dimension (see Fig. 8(g)).

5.1.1.6. Part-based vs. holistic approaches. Holistic methods [2,17,43,39,37] formulate the detection problem globally. Linemod [2] represents the windows extracted from RGB and depth images by the surface normals and color gradients features. Distortions along the object borders arising from occlusion and clutter, that is, the distortions of the color gradient and surface normal information in the test processes, mainly degrade the performance of this detector. Part-based methods [4,28,35,30] extract parts in the given image. Despite the fact that LCHF uses the same kinds of features as in Linemod, LCHF detects objects extracting parts, thus making the method more robust to occlusion and clutter.

5.1.1.7. Template-based vs. random forest-based. Template-based methods, *i.e.*, Linemod, match the features extracted during test to a set of templates, and hence, they cannot easily be generalized well to

unseen ground truth annotations, that is, the translation and rotation parameters in object pose estimation. Methods based on random forests [4,28,30] efficiently benefit the randomization embedded in this learning tool, consequently providing good generalization performance on new unseen samples.

5.1.1.8. RGB-D vs. depth. Methods utilizing both RGB and depth channels demonstrate higher recall values than methods that are of using only depth, since RGB provides extra clues to ease the detection. This is depicted in Table 8(a) where learning- and template-based methods of RGB-D perform much better than point-to-point technique [17] of depth channel.

5.1.1.9. RGB-D vs. RGB (CNN structures). More recent paradigm is to adopt CNNs to solve 6D object pose estimation problem taking RGB images as inputs [37,39]. The methods [37,38,40] working in the RGB channel in Table 8 are based on CNN structure. According to the numbers presented in Table 8, RGB-based SSD-6D [37] and RGB-D-based LCHF achieve similar performance. These recall values show the
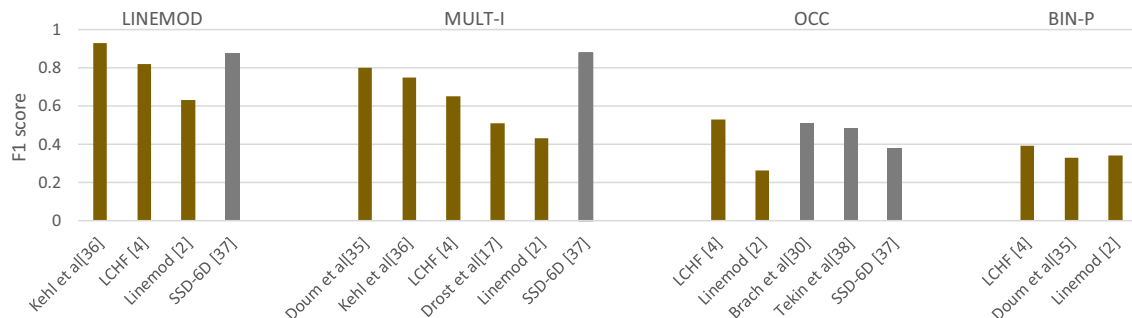
Fig. 9. Success of each baseline on every dataset is shown, F1 scores are computed using the Average Distance (AD) metric.
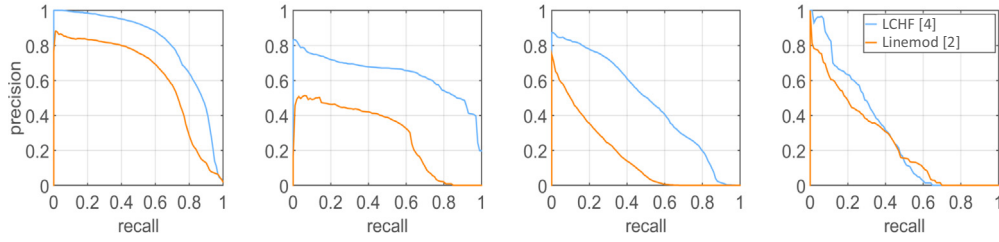
Fig. 10. Precision-recall curves of averaged F1 scores for LCHF [4] and Linemod [2] are shown: from left to right, LINEMOD, MULT-I, OCC, BIN-P.

promising performance of CNN architectures across random forest-based learning methods.

Robotic manipulators that pick and place the items from conveyors, shelves, pallets, *etc.*, need to know the pose of one item per RGB-D image, even though there might be multiple items in its workspace. Hence our recall-only analyses mainly target to solve the problems that could be encountered in such cases. Based upon the analyses currently made, one can make important implications, particularly from the point of the performances of the detectors. On the other hand, recall-based analyses are not enough to illustrate which dataset is more challenging than the others. This is especially true in crowded scenarios where multiple instances of target objects are severely occluded and cluttered. Therefore, in the next part, we discuss the performances of the baselines from another aspect, regarding precision-recall curves and F1 scores, where the 6D detectors are investigated sorting all detection scores across all images.

5.1.2. Precision-recall discussions

Our precision-recall discussions are based on the F1 scores provided in Table 9, and Fig. 9.

We first analyze the performance of the methods [36,4,2,37] on the LINEMOD dataset. On the average, Kehl et al. [36] outperforms other methods proving the superiority of learning deep features. Despite estimating 6D in RGB images, SSD-6D [37] exhibits the advantages of using CNN structures for 6D object pose estimation. LCHF and Linemod demonstrate lower performance, since the features used by these methods are manually-crafted. The comparison between Figs. 7 and 9 reveals that the results produced by the methods have approximately the same characteristics on the LINEMOD dataset, with respect to recall and F1 scores.

The methods tested on the MULT-I dataset [35,36,4,17,2] utilize the geometry information inherently provided by depth images. Despite this fact, SSD-6D [37], estimating 6D pose only from RGB images, outperforms other methods clearly proving the superiority of using CNNs for the 6D problem over other structures.

LCHF [4] and Brachmann et al. [30] best perform on OCC with respect to F1 scores. As this dataset involves test images where highly occluded objects are located, the reported results depict the importance of designing part-based solutions.

The most important difference is observed on the BIN-P dataset. While the success rates of the detectors on this dataset are higher than 60% with respect to the recall values (see Fig. 7), according to the presented F1 scores, their performance are less than 40%. When we take into account all hypotheses and the challenges particular to this dataset, which are severe occlusion and severe clutter, we observe strong degradation in the accuracy of the detectors.

In Fig. 10, we lastly report precision-recall curves of LCHF and Linemod. Regarding these curves, one can observe that as the datasets are getting more difficult, from the point of challenges involved, the methods produce less accurate results.

5.2. Analyses based on Visible Surface Discrepancy

The analyses presented so far have been employed using the AD metric. We continue our discussions computing recall values for the two methods LCHF [4] and Linemod [2], using the VSD metric. VSD renders the model of an object of interest at both ground truth and estimated poses, and intersects the depth maps of renderings with the test image in order to compute visibility masks. In the calculation of the visibility masks, there are several important parameters: i) the
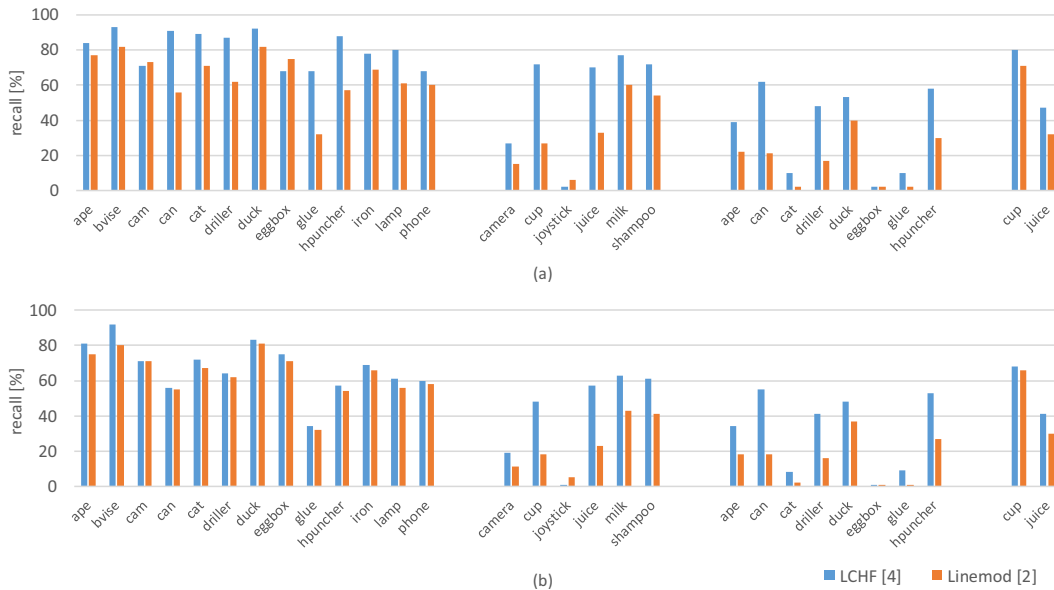

Fig. 11. Baselines are compared using Visible Surface Discrepancy (VSD) with respect to recall values. The performances of the baselines are depicted object-wise: from left to right, LINEMOD, MULT-I, OCC, and BIN-P. Parameters: (a) $\delta = 20$, $\tau = 100$, and $t = 0.5$, (b) $\delta = 15$, $\tau = 100$, and $t = 0.5$.
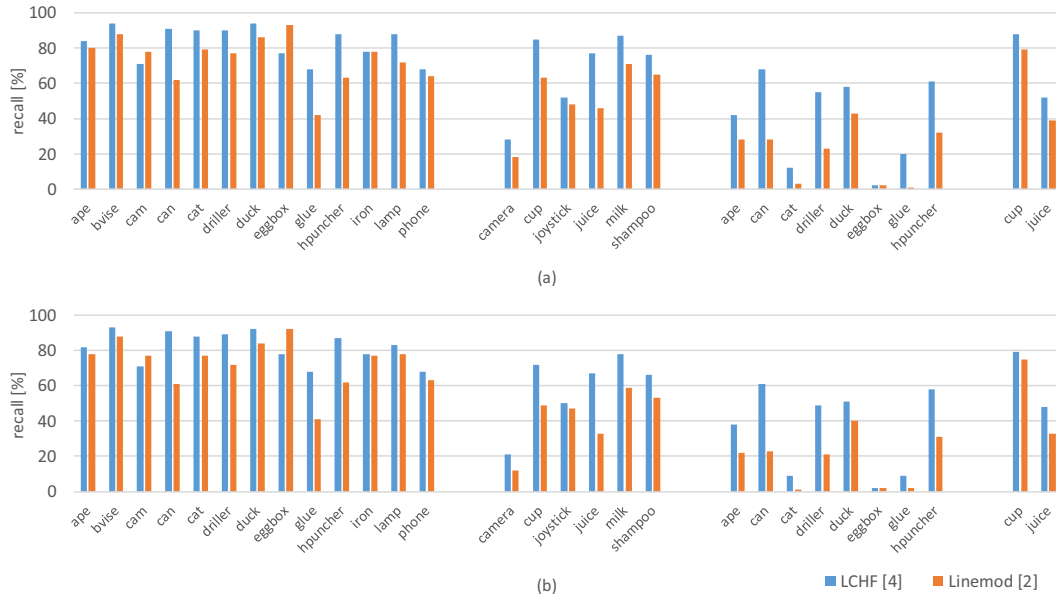
Fig. 12. Baselines are compared using Visible Surface Discrepancy (VSD) with respect to recall values. The performances of the baselines are depicted object-wise: from left to right, LINEMOD, MULT-I, OCC, and BIN-P. Parameters: (a) $\delta = 20$, $\tau = 100$, and $t = 0.6$, (b) $\delta = 15$, $\tau = 100$, and $t = 0.6$.

tolerance δ, ii) the misalignment tolerance τ, and iii) the threshold t. We firstly set $\delta = 20$ mm, $\tau = 100$ mm, and $t = 0.5$, and the bar chart in Fig. 11(a) demonstrates the object-wise accuracy of each baseline on the LINEMOD, MULT-I, OCC, BIN-P datasets, respectively. Both baselines demonstrate over 60% recall on more than half of the objects involved in the LINEMOD dataset. The object glue is the one on which the methods degrade. In the MULT-I dataset, part-based LCHF performs better than Linemod. However, LCHF fails on detecting joystick, whilst the Linemod detector shows approximately 7% recall on this object. The baselines underperform on OCC, since this dataset has the test images where the objects are subjected to severe occlusion. The objects cat, eggbox, and glue are the most difficult ones, for which the performances of the methods are limited to at most 10% recall. Apart from severe occlusion, eggbox and glue have planar surfaces, an extra factor that impairs the features extracted in depth channel. Both baselines produce higher recall values on the cup object involved in the BIN-P dataset, whilst the methods' performance decreases when the concerned object is juice, since multiple planar surfaces in the scene cause severe clutter for the detectors.

Next, we present the accuracy of each method altering the parameters of VSD. Fig. 11(b) shows the recall values when $\delta = 15$, $\tau = 100$, and $t = 0.5$. Note that these are also the parameters used in the experiments of the original publication of VSD [3]. Slightly decreasing the parameter δ gives rise lower recall values, however, we cannot see any significant difference from the point of characteristics of the bar charts. Unlike the parameter δ, when we change the threshold t, one can observe remarkable discrepancies for several objects. Fig. 12 demonstrates

the accuracy of each method when the parameters δ, τ, t are set to 20 mm, 100 mm, 0.6, and 15 mm, 100 mm, 0.6, respectively. The comparison between Figs. 11 and 12 depicts that the moderation in the parameter t gives rise approximately 50% accuracy on the joystick object for both LCHF and Linemod.

We lastly average the object-wise results shown in Figs. 11 and 12, and report the success of each baseline for each dataset in Fig. 13. According to this figure, highest recall values are reached when $\delta = 20$, $\tau = 100$, and $t = 0.6$. According to the visual results, we found that the set of parameters, $\delta = 20$, $\tau = 100$, and $t = 0.5$, is the one most appropriate.

In Fig. 14, the evaluations of LCHF and Linemod are compared with respect to both AD ($z_\omega = 0.1$) and VSD ($\delta = 20$, $\tau = 100$, and $t = 0.5$). The characteristic of both chart is the same, that is, both methods, according to AD and VSD, perform best on the LINEMOD dataset, whilst worst on OCC. On the other hand, the main advantage of VSD is that it features ambiguity-invariance: since it is designed to evaluate the baselines over the visible parts of the objects, it gives more robust measurements across symmetric objects. Sample images in Fig. 14 show the hypotheses of symmetric objects which are considered as false positive according to the AD metric, whilst VSD accepts those as correct.

6. Summary and observations

Review studies available in the literature introduced important implications for the problem of object pose recovery, however, they
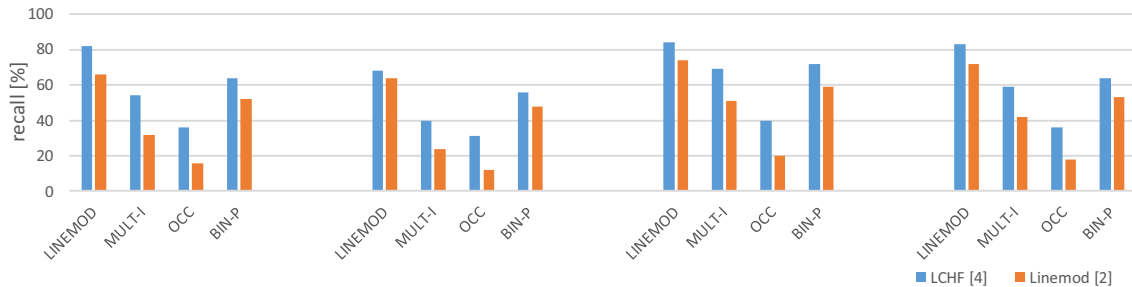

Fig. 13. Success of each baseline on every dataset is demonstrated averaging the recall values of individual objects. (a) $\delta = 20$, $\tau = 100$, $t = 0.5$, (b) $\delta = 15$, $\tau = 100$, $t = 0.5$, (c) $\delta = 20$, $\tau = 100$, $t = 0.6$, (d) $\delta = 15$, $\tau = 100$, $t = 0.6$.
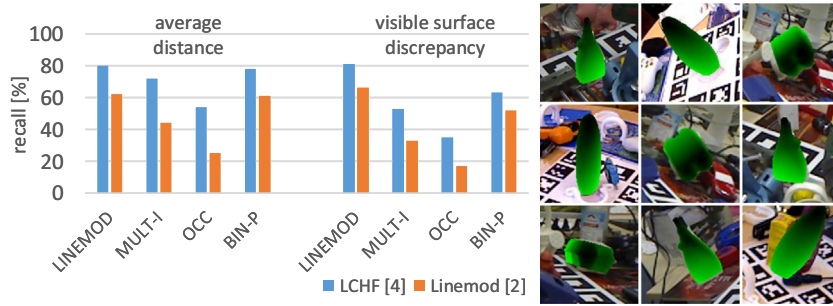
Fig. 14. Evaluations of LCHF and Linemod are compared with respect to both AD ($z_\omega = 0.1$) and VSD ($\delta = 20$, $\tau = 100$, and $t = 0.5$). Samples on the right are considered as false positive with respect to AD, whilst VSD deems correct.

mainly addressed generalized object detection in which the reviewed methods and discussions were restricted to visual level in 2D. In this paper, we have taken the problem a step further and have presented a comprehensive review of the methods ranging from 3D BB detectors to full 6D pose estimators. We have formed a discriminative categorization of the methods based on their problem modeling approaches. Along with the examination on the datasets and evaluations metrics, we have conducted multi-modal analyses investigating methods' performance in RGB and depth channels.

2D-driven 3D BB detection methods enlarge the 2D search space using the available appearance and geometry information in the 3D space along with RGB images. The methods directly detecting 3D BBs either use monocular RGB images together with contextual models as well as semantics or utilize stereo images, RGB-D cameras, and LIDAR sensors. On one hand, RGB provides dense measurements, however it is not efficient for handling objects located far away. On the other hand, LIDAR sensors precisely calculate depth information while its downside is the sparsity. Capitalizing their advantages, recent methods fuse two channels.

Template matching, point-pair feature matching, and random forest-based learning methods for 6D object pose estimation have dominated the area within the first half of the past decade. Typically working in RGB-D or in depth channel, accurate results have been obtained on textured-objects at varying viewpoints with cluttered backgrounds. Part-based solutions can handle the occlusion problem better than the ones global, using the information acquired from occlusion-free parts of the target objects. In the BOP challenge, point-pair feature matching methods have shown overperformance, however, heavy existence of occlusion and clutter severely affects the detectors. Similar-looking distractors along with similar-looking object classes seem the biggest challenge for such methods, since the lack of discriminative selection of shape features strongly confuse recognition systems. Hence, the community has initiated learning deep discriminative features. The most recent paradigm is to adopt end-to-end trainable CNN architectures, and solve the problem in RGB-only.

As 3D amodal BB detection methods are engineered to work at the level of categories, they are mainly evaluated on the KITTI [110], NYU-Depth v2 [111], and SUN RGB-D [112] datasets, where the objects of interest are aligned with the gravity direction. Unlike the 3D methods, full 6D object pose estimators are typically designed to work at the level of instances where the models of the objects of interest are utilized. 6D methods are tested on the datasets, such as LINEMOD [2], MULT-I [4], T-LESS [42], which have 6D object pose annotations. Exploiting the advantages of both category- and instance-level methods, 6D object pose estimation could be addressed at the level of categories. To this end, the random forest-based ISA [113] utilizes skeleton structures of models along with Semantically Selected Centers. Recently presented deep net in [114] estimates Normalized Object Coordinate Space, and has no dependence on the objects' 3D models. These methods could further be improved explicitly handling the challenges of the categories, *e.g.*, intra-class variations, distribution shifts. The creation of novel 6D

annotated datasets for categories can allow the community to measure the robustness of the methods.

Declaration of competing interest

We do not have any conflict of interest.

References

[1] C. Eppner, S. Hofer, R. Jonschkowski, R. Martin-Martin, A. Sieverling, V. Wall, O. Brock, Lessons from the Amazon Picking Challenge: four aspects of building robotic systems, Proceedings of Robotics: Science and Systems, 2016.

[2] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model Based Training, Detection, and Pose Estimation of Texture-less 3D Objects in Heavily Cluttered Scenes, ACCV 2012.

[3] T. Hodan, J. Matas, S. Obdrzalek, On Evaluation of 6D Object Pose Estimation, ECCVW 2016.

[4] A. Tejani, D. Tang, R. Kouskouridas, T.-K. Kim, Latent-class Hough Forests for 3D Object Detection and Pose Estimation, ECCV 2014.

[5] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, C. Rother, BOP: Benchmark for 6D Object Pose Estimation, ECCV 2018.

[6] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-scale Hierarchical Image Database, CVPR 2009.

[7] M. Everingham, L.V. Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, IJCV 2010.

[8] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, M. Hebert, An Empirical Study of Context in Object Detection, CVPR 2009.

[9] D. Hoiem, Y. Chodpathumwan, Q. Dai, Diagnosing Error in Object Detectors, ECCV 2012.

[10] O. Russakovsky, J. Deng, Z. Huang, A.C. Berg, L. Fei-Fei, Detecting Avocados to Zucchinis: What Have We Done, and Where Are We Going?, ICCV 2013.

[11] A. Torralba, A.A. Efros, Unbiased Look at Dataset Bias, CVPR 2011.

[12] M. Everingham, S.A. Eslami, L.V. Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge: A Retrospective, IJCV 2015.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, ImageNet Large Scale Visual Recognition Challenge, IJCV 2015.

[14] R.B. Rusu, N. Blodow, M. Beetz, Fast Point Feature Histograms (FPFH) for 3D Registration, ICRA 2009.

[15] E. Kim, G. Medioni, 3D Object Recognition in Range Images Using Visibility Context, IROS 2011.

[16] C. Papazov, D. Burschka, An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes, Asian Conference on Computer Vision 2010.

[17] B. Drost, M. Ulrich, N. Navab, S. Ilic, Model Globally, Match Locally: Efficient and Robust 3D Object Recognition, CVPR 2010.

[18] C. Choi, H.I. Christensen, 3D Pose Estimation of Daily Objects Using an RGB-D Camera, IROS 2012.

[19] C. Choi, Y. Taguchi, O. Tuzel, M.L. Liu, S. Ramalingam, Voting-based Pose Estimation for Robotic Assembly Using a 3D Sensor, International Conference on Robotics and Automation 2012.

[20] B. Drost, S. Ilic, 3D object detection and localization using multimodal point pair features, 3D Imaging, Modeling, Processing, Visualization and Transmission, 2012.

[21] R.P. de Figueiredo, P. Moreno, A. Bernardino, Fast 3D object recognition of rotationally symmetric objects, Pattern Recognition and Image Analysis, 2013.

[22] O. Tuzel, M.Y. Liu, Y. Taguchi, A. Raghunathan, Learning to Rank 3D Features, ECCV 2014.

[23] M.Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T.K. Marks, R. Chellappa, Fast Object Localization and Pose Estimation in Heavy Clutter for Robotic Bin Picking, IJRR 2012.

[24] R. Rios-Cabrera, T. Tuytelaars, Discriminatively Trained Templates for 3D Object Detection: A Real Time Scalable Approach, ICCV 2013.

[25] T. Hodan, X. Zabulis, M. Lourakis, S. Obdrzalek, J. Matas, Detection and Fine 3D Pose Estimation of Texture-less Objects in RGB-D Images, IROS 2015.

[26] J. Sock, S.H. Kasaei, L.S. Lopes, T.-K. Kim, Multi-view 6D object pose estimation and camera motion planning using RGBD images, 3rd International Workshop on Recovering 6D Object Pose, 2017.

[27] U. Bonde, V. Badrinarayanan, R. Cipolla, Robust Instance Recognition in Presence of Occlusion and Clutter, ECCV 2014.

[28] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, C. Rother, Learning 6D Object Pose Estimation Using 3D Object Coordinates, ECCV 2014.

[29] A. Krull, E. Brachmann, F. Michel, M.Y. Yang, S. Gumhold, C. Rother, Learning Analysis-by-synthesis for 6D Pose Estimation in RGB-D Images, ICCV 2015.

[30] E. Brachmann, F. Michel, A. Krull, M.Y. Yang, S. Gumhold, C. Rother, Uncertainty-driven 6D Pose Estimation of Objects and Scenes From a Single RGB Image, CVPR 2016.

[31] C. Sahin, R. Kouskouridas, T.-K. Kim, Iterative Hough Forest With Histogram of Control Points for 6 DoF Object Registration From Depth Images, IROS 2016.

[32] C. Sahin, R. Kouskouridas, T.-K. Kim, A learning-based variable size part extraction architecture for 6D object pose recovery in depth images, Journal of Image and Vision Computing 63 (2017) 38–50.

[33] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, C. Rother, Global Hypothesis Generation for 6D Object Pose Estimation, CVPR 2017.

[34] P. Wohlhart, V. Lepetit, Learning Descriptors for Object Recognition and 3D Pose Estimation, CVPR 2015.

[35] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, T.-K. Kim, Recovering 6D Object Pose and Predicting Next-Best-view in the Crowd, CVPR 2016.

[36] W. Kehl, F. Milletari, F. Tombari, S. Ilic, N. Navab, Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation, ECCV 2016.

[37] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, N. Navab, SSD-6D: Making RGB-based 3D Detection and 6D Pose Estimation Great Again, CVPR 2017.

[38] B. Tekin, S.N. Sinha, P. Fua, Real-Time Seamless Single Shot 6D Object Pose Prediction, CVPR 2018.

[39] Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes, Robotics: Science and Systems (RSS), 2018.

[40] M. Rad, V. Lepetit, BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth, ICCV 2017.

[41] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, T.-K. Kim, Pose Guided RGBD Feature Learning for 3D Object Pose Estimation, ICCV 2017.

[42] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, X. Zabulis, T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects, WACV 2017.

[43] S. Hinterstoisser, V. Lepetit, N. Rajkumar, K. Konolige, Going Further with Point-pair Features, ECCV 2016.

[44] W. Kehl, F. Tombari, N. Navab, S. Ilic, V. Lepetit, Hashmod: A Hashing Method for Scalable 3D Object Detection, BMVC 2015.

[45] A. Hettiarachchi, D. Wigdor, Annexing reality: enabling opportunistic use of everyday objects as tangible proxies in augmented reality, Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM 2016, pp. 1957–1967.

[46] D. Lindlbauer, J. Mueller, M. Alexa, Changing the appearance of real-world objects by modifying their surroundings, Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM 2017, pp. 3954–3965.

[47] F. Born, M. Masuch, Increasing presence in a mixed reality application by integrating a real time tracked full body representation, International Conference on Advances in Computer Entertainment, Springer, Cham 2017, pp. 46–60.

[48] R. Xiao, J. Schwarz, N. Throm, A.D. Wilson, H. Benko, MRTouch: adding touch input to head-mounted mixed reality, IEEE Trans. Vis. Comput. Graph. 24 (4) (2018) 1653–1660.

[49] M. Sra, S. Garrido-Jurado, C. Schmandt, P. Maes, Procedurally generated virtual reality from 3D reconstructed physical space, Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, ACM 2016, pp. 191–200.

[50] S.-Y. Teng, T.-S. Kuo, C. Wang, C.-H. Chiang, D.-Y. Huang, L. Chan, B.-Y. Chen, PuPoP: Pop-up Prop on Palm for virtual reality, The 31st Annual ACM Symposium on User Interface Software and Technology, ACM 2018, pp. 5–17.

[51] D. Lindlbauer, A.D. Wilson, Remixed reality: manipulating space and time in augmented reality, Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ACM 2018, p. 129.

[52] D. Harley, A.P. Tarun, D. Germinario, A. Mazalek, Tangible VR: diegetic tangible objects for virtual reality narratives, Proceedings of the 2017 Conference on Designing Interactive Systems, ACM 2017, pp. 1253–1263.

[53] I.F. Mondragon, P. Campoy, C. Martinez, M.A. Olivares-Mendez, 3D Pose Estimation Based on Planar Object Tracking for UAVs Control, ICRA 2010.

[54] I. Dryanovski, W. Morris, J. Xiao, An Open-source Pose Estimation System for Micro-air Vehicles, ICRA 2011.

[55] A. Causo, Z.H. Chong, R. Luxman, Y.Y. Kok, Z. Yi, W.C. Pang, R. Meixuan, Y.S. Teoh, W. Jing, H.S. Tju, I.M. Chen, A Robust Robot Design for Item Picking, ICRA 2018.

[56] J. Huang, M. Cakmak, Code3: a system for end-to-end programming of mobile manipulator robots for novices and experts, Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction ACM, 2017.

[57] R. Patel, R. Curtis, B. Romero, N. Correll, Improving grasp performance using in-hand proximity and contact sensing, Robotic Grasping and Manipulation Challenge 2016, pp. 146–160.

[58] M. Schwarz, A. Milan, A.S. Periyasamy, S. Behnke, RGB-D object detection and semantic segmentation for autonomous manipulation in clutter, The International Journal of Robotics Research 37 (4–5) (2018) 437–451.

[59] A. Zeng, S. Song, K.T. Yu, E. Donlon, F.R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, Robotic Pick-and-place of Novel Objects in Clutter with Multi-affordance Grasping and Cross-domain Image Matching, ICRA 2018.

[60] M. Zhu, K.G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, K. Daniilidis, Single Image 3D Object Detection and Pose Estimation for Grasping, ICRA 2014.

[61] M. Blösch, S. Weiss, D. Scaramuzza, R. Siegwart, Vision Based MAV Navigation in Unknown and Unstructured Environments, ICRA 2010.

[62] C. Sahin, T.K. Kim, Recovering 6D object pose: A review and multi-modal analysis, ECCVW on ACVR, ECCV Workshops, 2018.

[63] E. Sucar, J.B. Hayet, Bayesian Scale Estimation for Monocular SLAM Based on Generic Object Detection for Correcting Scale Drift, ICRA 2018.

[64] K. Tateno, F. Tombari, I. Laina, N. Navab, CNN-SLAM: Real-time Dense Monocular SLAM With Learned Depth Prediction, CVPR 2017.

[65] P. Parkhiya, R. Khawad, J.K. Murthy, B. Bhowmick, K.M. Krishna, Constructing Category-specific Models for Monocular Object-SLAM, ICRA 2018.

[66] J. McCormac, R. Clark, M. Bloesch, A. Davison, S. Leutenegger, Fusion++: volumetric object-level SLAM, International Conference on 3D Vision (3DV) 2018, pp. 32–41.

[67] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, Z. Liu, H.I. Christensen, F. Dellaert, Multi robot object-based SLAM, International Symposium on Experimental Robotics, Springer 2016, pp. 729–741.

[68] M. Hosseinzadeh, K. Li, Y. Latif, I. Reid, Real-Time Monocular Object-Model Aware Sparse SLAM, ICRA 2019.

[69] D. Gálvez-López, M. Salas, J.D. Tardós, J.M.M. Montiel, Real-time monocular object SLAM, Robot. Auton. Syst. 75 (2016) 435–449.

[70] B. Mu, S.Y. Liu, L. Paull, J. Leonard, J.P. How, SLAM With Objects Using a Nonparametric Pose Graph, IROS 2016.

[71] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H. Kelly, A.J. Davison, SLAM++: Simultaneous Localisation and Mapping at the Level of Objects, CVPR 2013.

[72] C. Sahin, M. Unel, Under vehicle perception for high level safety measures using a catadioptric camera system, IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society, IEEE 2013, pp. 4306–4311.

[73] C. Zach, A. Penate-Sanchez, M.T. Pham, A Dynamic Programming Approach for Fast and Robust Object Pose Recognition From Range Images, CVPR 2015.

[74] J. Sock, K.I. Kim, C. Sahin, T.-K. Kim, Multi-Task Deep Networks for Depth-Based 6D Object Pose and Joint Registration in Crowd Scenarios, BMVC 2018.

[75] D. Lin, S. Fidler, R. Urtasun, Holistic Scene Understanding for 3D Object Detection with RGBD Cameras, ICCV 2013.

[76] S. Gupta, R. Girshick, P. Arbelaez, J. Malik, Learning Rich Features from RGB-D Images for Object Detection and Segmentation, ECCV 2014.

[77] M. Engelcke, D. Rao, D.Z. Wang, C.H. Tong, I. Posner, Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks, ICRA 2017.

[78] Y. Zhang, M. Bai, P. Kohli, S. Izadi, J. Xiao DeepContext, Context-Encoding Neural Pathways for 3D Holistic Scene Understanding, ICCV 2017.

[79] S. Song, J. Xiao, Deep Sliding Shapes for Amodal 3D object Detection in RGB-D Images, CVPR 2016.

[80] Y. Zhou, O. Tuzel, VoxelNet: End-to-end Learning for Point Cloud Based 3D Object Detection, CVPR 2018.

[81] S. Gupta, P. Arbelaez, R. Girshick, J. Malik, Aligning 3D models to RGB-D Images of Cluttered Scenes, CVPR 2015.

[82] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3D Object Detection Network for Autonomous Driving, CVPR 2017.

[83] J. Lahoud, B. Ghanem, 2D-driven 3D Object Detection in RGB-D Images, ICCV 2017.

[84] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, R. Urtasun, Monocular 3D object detection for autonomous driving, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[85] Z. Deng, L. Jan Latecki, Amodal detection of 3D objects: Inferring 3D bounding boxes from 2D ones in RGB-depth images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 5762–5770.

[86] S. Gupta, P. Arbelaez, J. Malik, Perceptual organization and recognition of indoor scenes from RGB-D images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013, pp. 564–571.

[87] A. Mousavian, D. Anguelov, J. Flynn, J. Kosecka, 3D bounding box estimation using deep learning and geometry, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 7074–7082.

[88] J. Papon, M. Schoeler, Semantic pose using deep networks trained on synthetic RGB-D, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 774–782.

[89] S. Song, J. Xiao, Sliding shapes for 3D object detection in depth images, European Conference on Computer Vision, Springer, Cham 2014, pp. 634–651.

[90] Y. Xiang, W. Choi, Y. Lin, S. Savarese, Data-driven 3D voxel patterns for object category recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 1903–1911.

[91] U. Asif, M. Bennamoun, F.A. Sohel, RGB-D object recognition and grasp detection using hierarchical cascaded forests, IEEE Trans. Robot. 33 (3) (2017) 547–564.

[92] M. Bleyer, C. Rhemann, C. Rother, Extracting 3D scene-consistent object proposals and depth from stereo images, European Conference on Computer Vision, Springer, Berlin, Heidelberg 2012, pp. 467–481.

[93] H. Jiang, Finding approximate convex shapes in RGBD images, European Conference on Computer Vision, Springer, Cham 2014, pp. 582–596.

[94] H. Jiang, J. Xiao, A linear approach to matching cuboids in RGBD images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013, pp. 2171–2178.

[95] S.H. Khan, X. He, M. Bennamoun, F. Sohel, R. Togneri, Separating objects and clutter in indoor scenes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 4603–4611.

[96] D.Z. Wang, I. Posner, Voting for voting in online point cloud object detection, Robotics: Science and Systems (Vol. 1, No. 3, Pp. 10–15607), 2015.

[97] R.B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, Towards 3D object maps for autonomous household robots, In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3191–3198). (IEEE).

[98] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, and M. Beetz, Towards 3D point cloud based object maps for household environments, Robotics and Autonomous Systems, 56(11), pp.927–941.

[99] D. Gehrig, P. Krauthausen, L. Rybok, H. Kuehne, U.D. Hanebeck, T. Schultz, and R. Stiefelhagen, Combined intention, activity, and motion recognition for a humanoid household robot, In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4819–4825). (IEEE).

[100] C.R. Qi, W. Liu, C. Wu, H. Su, L.J. Guibas, Frustum pointnets for 3D object detection from RGB-D data, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 918–927.

[101] Y. Wang, W.L. Chao, D. Garg, B. Hariharan, M. Campbell, K.Q. Weinberger, Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 8445–8453.

[102] S. Shi, X. Wang, H. Li, PointRCNN: 3D object proposal generation and detection from point cloud, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 770–779.

[103] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, PointPillars: fast encoders for object detection from point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 12697–12705.

[104] B. Li, W. Ouyang, L. Sheng, X. Zeng, X. Wang, GS3D: an efficient 3D object detection framework for autonomous driving, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 1019–1028.

[105] X. Chen, K. Kundu, Y. Zhu, A.G. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3D object proposals for accurate object class detection, Advances in Neural Information Processing Systems 2015, pp. 424–432.

[106] J. Ku, A.D. Pon, S.L. Waslander, Monocular 3D object detection leveraging accurate proposals and shape reconstruction, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 11867–11876.

[107] M. Liang, B. Yang, Y. Chen, R. Hu, R. Urtasun, Multi-task multi-sensor fusion for 3D object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 7345–7353.

[108] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S.L. Waslander, Joint 3D proposal generation and object detection from view aggregation, In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1–8). (IEEE).

[109] Z.Q. Zhao, P. Zheng, S.T. Xu, X. Wu, Object detection with deep learning: a review, IEEE Transactions on Neural Networks and Learning Systems, 2019.

[110] A. Geiger, P. Lenz, and R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354–3361). (IEEE).

[111] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, Indoor segmentation and support inference from RGBD images, In European Conference on Computer Vision (pp. 746–760). Springer, Berlin, Heidelberg.

[112] S. Song, S. Lichtenberg, J. Xiao, SUN RGB-D: A RGBD Scene Understanding Benchmark Suite, In CVPR 2015.

[113] C. Sahin, T.-K. Kim, Category-level 6D Object Pose Recovery in Depth Images, ECCVW on R6D, 2018.

[114] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, L.J. Guibas, Normalized object coordinate space for category-level 6D object pose and size estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 2642–2651.

[115] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 580–587.

[116] R. Girshick, Fast R-CNN, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 1440–1448.

[117] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems 2015, pp. 91–99.

[118] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, Advances in Neural Information Processing Systems 2016, pp. 379–387.

[119] T.Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 2117–2125.

[120] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 779–788.

[121] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single shot multibox detector, European Conference on Computer Vision, Springer, Cham 2016, pp. 21–37.

[122] P. Krahenbuhl, V. Koltun, Geodesic object proposals, European Conference on Computer Vision, Springer, Cham 2014, pp. 725–739.

[123] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, European Conference on Computer Vision, Springer, Cham 2016, pp. 354–370.

[124] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[125] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems 2012, pp. 1097–1105.

[126] X. Ren, D. Ramanan, Histograms of Sparse Codes for Object Detection, CVPR 2013.

[127] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[128] J. Gall, V. Lempitsky, Class-specific Hough forests for object detection, Decision Forests for Computer Vision and Medical Image Analysis, 2013.

[129] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, Adv. Neural Inf. Proces. Syst. (2007) 985–992.

[130] U. Bonde, T.-K. Kim, K.R. Ramakrishnan, Randomised manifold forests for principal angle-based face recognition, Asian Conference on Computer Vision, 2010.

[131] C. Leistner, A. Saffari, J. Santner, H. Bischof, Semi-supervised Random Forests, ICCV 2009.

[132] M. Sun, P. Kohli, J. Shotton, Conditional Regression Forests for Human Pose Estimation, CVPR 2012.

[133] J. Gall, A. Yao, N. Razavi, L.V. Gool, V. Lempitsky, Hough Forests for Object Detection, Tracking, and Action Recognition, TPAMI 2011.

[134] A. Criminisi, J. Shotton, Decision Forests for Computer Vision and Medical Image Analysis, Springer Science and Business Media, 2013.

[135] T. Birdal, S. Ilic, Point pair features based object detection and pose estimation revisited, International Conference on 3D Vision. IEEE 2015, pp. 527–535.

[136] C. Choi, A.J. Trevor, H.I. Christensen, RGB-D edge detection and edge-based registration, IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE 2013, pp. 1568–1575.

[137] C. Choi, H.I. Christensen, RGB-D object pose estimation in unstructured environments, Rob. Auton. Syst. 75 (2016) 595–613.

[138] J. Carreira, C. Sminchisescu, CPMC: automatic object segmentation using constrained parametric min-cuts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2011) 1312–1328.

[139] P. Arbelaez, J. Pont-Tuset, J.T. Barron, F. Marques, J. Malik, Multi-scale combinatorial grouping, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 328–335.

[140] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3D classification and segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 652–660.

[141] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, Advances in Neural Information Processing Systems 2017, pp. 5099–5108.

[142] B. Li, 3D fully convolutional network for vehicle detection in point cloud, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017, pp. 1513–1518 , IEEE.

[143] M. Liang, B. Yang, S. Wang, R. Urtasun, Deep continuous fusion for multi-sensor 3D object detection, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 641–656.

[144] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, T. Chateau, Deep MANTA: a coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 2040–2049.

[145] B. Xu, Z. Chen, Multi-level fusion based 3D object detection from monocular images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 2345–2353.

[146] B. Yang, W. Luo, R. Urtasun, PIXOR: real-time 3D object detection from point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 7652–7660.

[147] D. Xu, D. Anguelov, A. Jain, PointFusion: deep sensor fusion for 3D bounding box estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 244–253.

[148] Y. Yan, Y. Mao, B. Li, SECOND: Sparsely embedded convolutional detection, Sensors 18 (10) (2018).

[149] Z. Ren, E.B. Sudderth, Three-dimensional object detection and layout prediction using clouds of oriented gradients, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 1525–1533.

[150] B. Li, T. Zhang, T. Xia, Vehicle detection from 3D lidar using fully convolutional network, Robotics: Science and Systems, 2016.

[151] Z. Wang, K. Jia, Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection, IEEE IROS 2019.

[152] D. Maturana, S. Scherer, VoxNet: a 3D convolutional neural network for real-time object recognition, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015, pp. 922–928 , IEEE.

[153] X. Du, M.H. Ang, S. Karaman, D. Rus, A general pipeline for 3D detection of vehicles, 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018, pp. 3194–3200 , IEEE.

[154] M.Z. Zia, M. Stark, K. Schindler, Are cars just 3D boxes? Jointly estimating the 3D shape of multiple objects, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 3678–3685.

[155] S. Song, M. Chandraker, Joint SFM and detection cues for monocular 3D localization in road scenes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 3734–3742.

[156] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-scale Image Recognition, ICLR 2015.

[157] A. Kanezaki, Y. Matsushita, Y. Nishida, RotationNet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 5010–5019.

[158] A. Kundu, Y. Li, J.M. Rehg, 3D-RCNN: instance-level 3D object reconstruction via render-and-compare, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 3559–3568.

[159] Y. Hu, J. Hugonot, P. Fua, M. Salzmann, Segmentation-driven 6D object pose estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 3385–3394.

[160] S. Peng, Y. Liu, Q. Huang, X. Zhou, H. Bao, PVNet: pixel-wise voting network for 6DoF pose estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 4561–4570.

[161] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, S. Savarese, DenseFusion: 6D object pose estimation by iterative dense fusion, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 3343–3352.

[162] C. Li, J. Bai, G.D. Hager, A unified framework for multi-view multi-class object pose estimation, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 254–269.

[163] M. Oberweger, M. Rad, V. Lepetit, Making deep heatmaps robust to partial occlusions for 3D object pose estimation, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 119–134.

[164] M. Sundermeyer, Z.C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3D orientation learning for 6D object detection from RGB images, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 699–715.

[165] R. Brégier, F. Devernay, L. Leyrit, J.L. Crowley, Symmetry aware evaluation of 3D object detection and pose estimation in scenes of many parts in bulk, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 2209–2218.

[166] S. Zakharov, I. Shugurov, and S. Ilic, DPOD: 6D Pose Object Detector and Refiner, ICCV19.

[167] Z. Li, G. Wang, and X. Ji, CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation, ICCV19.

[168] K. Park, T. Patten, and M. Vincze, Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation, ICCV19.

[169] A. Zeng, K.T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, J. Xiao, Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge, IEEE International Conference on Robotics and Automation (ICRA), 2017 , (pp. 1386-1383). IEEE.

[170] M. Sundermeyer, Z.C. Marton, M. Durner, R. Triebel, Augmented autoencoders: implicit 3D orientation learning for 6D object detection, International Journal of Computer Vision (2019) 1–16.

[171] J. Vidal, C.Y. Lin, X. Lladó, R. Martí, A method for 6D pose estimation of free-form rigid objects using point pair features on range data, Sensors 18 (8) (2018).

[172] P. Castro, A. Armagan, T.-K. Kim, Accurate 6D Object Pose Estimation by Pose Conditioned Mesh Reconstruction, arXiv preprint, 2019. arXiv:1910.10653.

[173] J. Sock, G. Garcia-Hernando, T.-K. Kim, Active 6D Multi-Object Pose Estimation in Cluttered Scenarios with Deep Reinforcement Learning, arXiv preprint, 2019. arXiv:1910.08811.

[174] A. Kendall, M. Grimes, R. Cipolla, PoseNet: a convolutional network for real-time 6-dof camera relocalization, The IEEE International Conference on Computer Vision (ICCV), December 2015.

[175] O. Erkent, D. Shukla, J. Piater, Integration of Probabilistic Pose Estimates from Multiple Views, In ECCV 2016.

[176] N. Payet, S. Todorovic, From contours to 3D object detection and pose estimation, International Conference on Computer Vision 2011, pp. 983–990 , IEEE.

[177] H. Tjaden, U. Schwanecke, E. Schomer, Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 124–132.

[178] Y. Konishi, Y. Hanzawa, M. Kawade, M. Hashimoto, Fast 6D pose estimation from a monocular image using hierarchical pose trees, European Conference on Computer Vision, Springer, Cham 2016, pp. 398–413.

[179] M. Ulrich, C. Wiedemann, C. Steger, Combining scale-space and similarity-based aspect graphs for fast 3D object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (10) (2011) 1902–1914.

[180] K. Park, T. Patten, J. Prankl, M. Vincze, Multi-task template matching for object detection, segmentation and pose estimation using depth images, International Conference on Robotics and Automation (ICRA) 2019, pp. 7207–7213 , IEEE.

[181] S. Zakharov, W. Kehl, B. Planche, A. Hutter, S. Ilic, 3D object instance recognition and pose estimation using triplet loss with dynamic margin, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017, pp. 552–559 , IEEE.

[182] E. Muñoz, Y. Konishi, V. Murino, A. Del Bue, Fast 6D pose estimation for texture-less objects from a single RGB image, 2016 IEEE International Conference on Robotics and Automation (ICRA) 2016, pp. 5623–5630 , IEEE.

[183] M. Li, K. Hashimoto, Accurate object pose estimation using depth only, Sensors 18 (4) (2018) 1045.

[184] C. Rennie, R. Shome, K.E. Bekris, A.F. De Souza, A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place, Robotics and Automation Letters, 2016.

[185] A. Janoch, S. Karayev, Y. Jia, J.T. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3-D object dataset: putting the kinect to work, ICCV Workshop on Consumer Depth Cameras in Computer Vision, 2011.

[186] J. Xiao, A. Owens, and A. Torralba, SUN3D: a database of big spaces reconstructed using SfM and object labels, Proceedings of 14th IEEE International Conference on Computer Vision (ICCV2013).

[187] C.R. Qi, O. Litany, K. He, L.J. Guibas, Deep Hough Voting for 3D Object Detection in Point Clouds, ICCV 2019.

[188] K. Lai, L. Bo, X. Ren, D. Fox, A Large-scale Hierarchical Multi-view RGB-D Object Dataset, In ICRA 2011.

[189] J. Lim, H. Pirsiavash, A. Torralba, Parsing IKEA Objects: Fine Pose Estimation, In ICCV 2013.

[190] K. Matzen, N. Snavely, NYC3DCars: A Dataset of 3D Vehicles in Geographic Context, In ICCV 2013.

[191] M. Ozuysal, V. Lepetit, P. Fua, Pose Estimation for Category Specific Multiview Object Localization, In CVPR 2009.

[192] M. Sun, G. Bradski, B.-X. Xu, S. Savarese, Depth-encoded Hough Voting for Joint Object Detection and Shape Recovery, In ECCV 2010.

[193] Y. Xiang, S. Savarese, Estimating the Aspect Layout of Object Categories, In CVPR 2012.

[194] Y. Xiang, R. Mottaghi, S. Savarese, Beyond PASCAL: a benchmark for 3D object detection in the wild, IEEE Winter Conference on Applications of Computer Vision 2014, pp. 75–82 , IEEE.

[195] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, A.M. Dollar, Yale-CMU-Berkeley dataset for robotic manipulation research, International Journal of Robotics Research, 2017.

[196] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, A. Dollar, The YCB object and model set: towards common benchmarks for manipulation research, International Conference on Advanced Robotics, 2015.

[197] C. Li, J. Bohren, E. Carlson, G.D. Hager, Hierarchical semantic parsing for object pose estimation in densely cluttered scenes, International Conference on Robotics Automation (ICRA), 2016.

[198] V. Hegde, R. Zadeh, Fusionnet: 3D Object Classification Using Multiple Data Representations, 3D Deep Learning Workshop at NIPS, 2016.

[199] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: a survey, Int. J. Comput. Vis. 1809 (2018) 1–58.

[200] D. Holz, M. Nieuwenhuisen, D. Droeschel, J. Stuckler, A. Berner, J. Li, R. Klein, S. Behnke, Active recognition and manipulation for mobile robot bin picking, Book Chapter in Gearing Up and Accelerating Cross-fertilization Between Academic and Industrial Robotics Research in Europe, Springer, 2014.

[201] M. Mohamad, D. Rappaport, M. Greenspan, Generalized 4-points congruent sets for 3D registration, IEEE International Conference on 3D Vision, 2014.

[202] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stuckler, A. Berner, J. Li, R. Klein, S. Behnke, Mobile bin picking with an anthropomorphic service robot, International Conference on Robotics and Automation, 2013.

[203] H. Deng, T. Birdal, S. Ilic, PPFNet: global context aware local features for robust 3D point matching, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 195–205.

[204] H. Deng, T. Birdal, S. Ilic, PPF-FoldNet: unsupervised learning of rotation invariant 3D local descriptors, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 602–618.

[205] T. Hodan, R. Kouskouridas, T.-K. Kim, F. Tombari, K. Bekris, B. Drost, T. Groueix, K. Walas, V. Lepetit, A. Leonardis, C. Steger, F. Michel, C. Sahin, C. Rother, J. Matas, A summary of the 4th International Workshop on Recovering 6D Object Pose, Computer Vision – ECCV 2018 Workshops, Springer, Cham Sept. 2018, pp. 589–600.

[206] C. Sahin, G. Garcia-Hernando, J. Sock, T.-K. Kim, Instance- and category-level 6D object pose estimation, Book Chapter in RGB-D Image Analysis and Processing, Springer 2019, pp. 243–265.