6D object position estimation from 2D images: a literature review

Multimedia Tools and Applications (2023) 82:24605–24643
https://doi.org/10.1007/s11042-022-14213-z

6D object position estimation from 2D images: a literature review

Giorgia Marullo[1] · Leonardo Tanzi[1] · Pietro Piazzolla[1] · Enrico Vezzetti[1]

Received: 20 May 2022 / Revised: 15 September 2022 / Accepted: 28 October 2022 /
Published online: 28 November 2022

Abstract
The 6D pose estimation of an object from an image is a central problem in many domains of Computer Vision (CV) and researchers have struggled with this issue for several years. Traditional pose estimation methods (1) leveraged on geometrical approaches, exploiting manually annotated local features, or (2) relied on 2D object representations from different points of view and their comparisons with the original image. The two methods mentioned above are also known as Feature-based and Template-based, respectively. With the diffusion of Deep Learning (DL), new Learning-based strategies have been introduced to achieve the 6D pose estimation, improving traditional methods by involving Convolutional Neural Networks (CNN). This review analyzed techniques belonging to different research fields and classified them into three main categories: Template-based methods, Feature-based methods, and Learning-Based methods. In recent years, the research mainly focused on Learning-based methods, which allow the training of a neural network tailored for a specific task. For this reason, most of the analyzed methods belong to this category, and they have been in turn classified into three sub-categories: Bounding box prediction and Perspective-n-Point (PnP) algorithm-based methods, Classification-based methods, and Regression-based methods. This review aims to provide a general overview of the latest 6D pose recovery methods to underline the pros and cons and highlight the best-performing techniques for each group. The main goal is to supply the readers with helpful guidelines for the implementation of performing applications even under challenging circumstances such as auto-occlusions, symmetries, occlusions between multiple objects, and bad lighting conditions.

Keywords Computer vision · 6D position estimation · Deep learning · RGB Input

✉ Enrico Vezzetti
enrico.vezzetti@polito.it

[1] Department of Management, Production and Design Engineering, Polytechnic University of Turin, Corso Duca degli Abruzzi, 24, 10129 Turin, Italy

1 Introduction

6D position estimation is an essential task in many Computer Vision (CV) applications. It concerns, among others, robotics [35], autonomous driving [20], and virtual/augmented reality (VR/AR) applications [27] and is extensively used in the entertainment and medical care industry [20]. The problem itself is simple and consists of determining the 3D rotation and translation of an object which shape is known in relation to the camera, using details observable from the reference 2D image. However, achieving a solution to this problem is not trivial [35]. Firstly, due to auto-occlusions or symmetries, the objects cannot be clearly and unequivocally identifiable. Moreover, the image conditions are not always optimal in term of lighting and occlusions between the objects represented in the picture [2, 20, 73]. In these situations, it is often necessary to add an earlier stage of object detection or localization to distinguish the area of the image which contains the object, before estimating its position.

Although the researchers have studied this problem for many years, it experienced a rebirth with the advent of Deep Learning (DL) [19], in the same way as other fields of application, such as the medical [52–54] or face recognition [38] domains. Old pose estimation methods were based on geometrical approaches, as for example Feature-based methods, which tried to establish correspondences between 3D models and 2D images of objects by using manually annotated local features. With texture-less or geometrically complex objects, it was not easy to select local features. In these cases, even though the matching phase usually took much time, it might fail and provide a result that was not always accurate [69].

In opposition to these methods, researchers introduced Template-based methods, which represented the 2D object from different points of view and compared these representations with the original image to establish the position and orientation. These approaches were very susceptible to variations in lighting and occlusions even if they could manage texture-less objects and required many comparisons to reach a certain accuracy level, increasing the execution time [27]. With the diffusion of DL, researchers improved traditional methods by introducing Learning-based methods, making them more efficient and performing. The basic idea of these systems involves Convolutional Neural Networks (CNN) to learn a mapping function between images with three-dimensional position annotations, and object 6D position. Some of these systems employ a CNN to predict the 2D projection of the 3D bounding box corners, and then the PnP algorithm. The PnP algorithm is extensively used in CV to calculate the 6D position from matches between 2D features on the test image and 3D points on the CAD model [16]. Other types of Learning-based methods, instead, need only a CNN to resolve a classification or a regression problem. For this reason, Learning-based methods are referred as Bounding box prediction and PnP algorithm-based, Classification-based, and Regression-based methods, respectively. These methods can reach very high levels of precision but need many data to train the network accurately and to be able to work well in real cases. Alternatively, CNNs can be used to execute the most critical steps of traditional methods to join the advantages of the various strategies into the final solution [69].

Referring to the methods mentioned above this literature review focused on the classification of 6D position estimation methods from a single RGB image. The main goal of this work is to supply a baseline for the development of new applications which can work even under boundary conditions, namely, auto-occlusions, symmetries, occlusions between multiple objects, and bad lighting conditions. These conditions, indeed, are widespread in real domains of application, for example, autonomous driving and the medical field. However, the literature review did not reveal a one-size-fits-all method for each case. Consequently, an attempt was

undertaken to establish guidelines for new applications, considering the context and related implementation conditions on the one hand and the availability of data and computing power on the other.

The paper is organized as follows: Section 2 describes the methodology used to select the articles, Section 3 illustrates Template-based methods, Section 4 describes Feature-based methods and Section 5 focuses on Learning-Based methods. These last methods have been in turn classified into three categories: Bounding box prediction and PnP algorithm-based methods (Section 5.1); Classification-based methods (Section 5.2); Regression-based methods (Section 5.3). Finally, Section 6 summarizes and discusses the study.

2 Methodology

In this literature review, the articles have been selected by involving different strategies to establish, among the hundreds of existing papers about the topic, the most suitable ones. In general, only papers published from 2016 on have been considered. A second discard criterion concerned input data, since only those systems having a single RGB image as input and optionally a 3D model of the object have been kept. Multiple or RGB-D cameras are hard to use, they are expensive, the calibration process becomes troublesome, and the equipment may become too heavy. As a result, articles including stereo images, RGB-D images, or data from sensors, such as LIDAR sensors, have been ignored, since they supplemented the input with depth information lessening the problem complexity.

Specifically, the articles have been searched on Google Scholar, IEEE, ACM Digital Library, Springer and Science Direct using as keywords "6D pose estimation from RGB images", "Viewpoint prediction", "Position and Orientation estimation", "3D point anchoring", "Automatic Registration". Indeed, these keywords appears to be consistent with the references of recent surveys on 6D pose estimation topic, such as the one proposed by Sahin et al. in [48]. From a first analysis, $n = 891$ articles have been selected, using as constraints: (1) Number of citations > 5; (2) Year of publication > 2015. These exclusion criteria have been chosen because, in general, articles with less than 5 citations were considered related to a narrower research field, i.e., they analyzed objects with peculiar features and, consequently, the proposed method was hardly generalizable to other contexts. However, although some recent papers did not satisfy the condition, there were included because they were considered relevant to our work. Furthermore, given the ongoing innovation of these technologies, this review focuses only on recent studies. In a second step, $n = 815$ studies have been excluded after screening titles and abstracts, obtaining $n = 76$ articles for a first full text review. Then, starting from the full text analyzed papers' bibliography, $n = 13$ new references have been obtained, reaching a total of $n = 89$ valuable articles for a second full text review. Finally, after this phase, $n = 32$ papers have been excluded for three main reasons: (1) $n = 14$ systems using RGB-D input, (2) $n = 12$ works concerning old technologies, (3) $n = 6$ papers which showed low results in terms of accuracy and, consequently, could not be exploited for applications related to critical research fields such as medicine.

Finally, although all the examined articles satisfied all the requirements, a further screening has been done, by considering the number of citations of each paper, according to the number reported by Google Scholar. The PRISMA [32] flowchart is shown in Fig. 1. Once selected, the articles were categorized according to the pipeline and architecture of the proposed method.

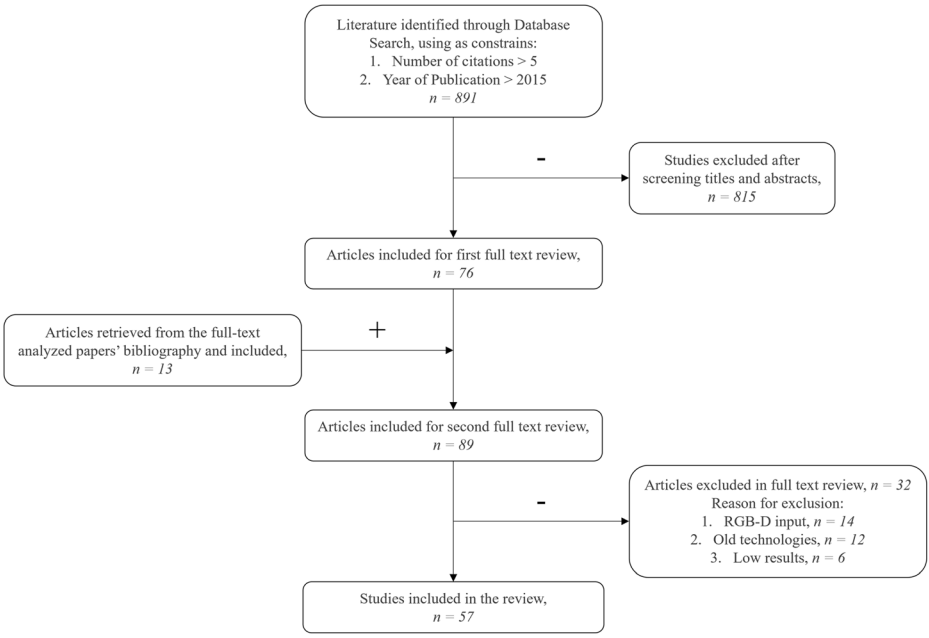

Fig. 1 PRISMA flowchart

Therefore, this review has three main features:

1. It considers only RGB images as input, excluding RGB-D images and data from LIDAR sensors because this information is often not available.
2. It first classified the selected articles into three main categories: feature-based, template-based, and learning-based methods.
3. Then, as the learning-based approaches are of great importance in research, it focuses on this class of methods and, in turn, categorizes them according to the task solved by the CNN: Bounding box prediction and PnP algorithm-based methods, where a CNN predicts the 3D bounding box, and then a PnP algorithm calculates the 6D position from matches between 2D features on the test image and 3D points on the CAD model [16]; Classification-based and Regression-based methods, in which the CNN resolves a classification or a regression problem, respectively.

To achieve the aim of the paper and provide a baseline for the development of new applications, the methods were analyzed considering specific parameters, as shown in the columns of the summary tables (Tables 1, 2, 3, 4 and 5). In addition to general information and the traditional selection criteria mentioned above, indeed, the following parameters were considered:

– *Pose Refinement*. Methods requiring an additional step to refine the estimated coordinates commonly need more processing time. Therefore, this characteristic is discerning in case an application needs to run in real-time.
– *Dataset*. Information regarding the dataset is an indicator of the ability of the system to be generalizable in different circumstances. For example, approaches that employ a dataset of only computer-generated images sometimes do not work satisfactorily with real images.

Table 1 Feature-based methods

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
Pavlakos et al. (2017)	2017	IEEE International Conference on Robotics and Automation (ICRA)	266	It combines semantic keypoints with a deformable shape model; it can manage scenes with cluttered backgrounds	Convnet with the stacked hourglass design for keypoints localization	RGB image	Bounding box detection by an off-the-shelf object detector	PnP algorithm
Peng et al. (2020)	2019	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	261	PVNet regresses pixel-wise unit vectors and uses them to vote for keypoint locations using RANSAC; it can manage severe occlusion or truncation; it can handle multiple instances	Pixel-wise Voting Network	RGB image	No	PnP algorithm
Zhu et al. (2021)	2021	Signal Processing: Image Communication	0	It tries to improve PVNet segmentation accuracy and the vector-field prediction by utilising the effective Atrous Spatial Pyramid Pooling module of Deeplabv3; it can handle severe occlusions	ASPP-PVNet	RGB image	No	PnP algorithm
You et al. (2021)	2021	IEEE Access	1	It considers a projection loss function dealing with the error of the vector field and incorporates a refinement network; it can simultaneously estimate the poses of multiple objects	ResNet-18 for extracting semantic segmentation and vector-field data	RGB image	No	PnP algorithm
Zhao et al. (2018)	2018	ArXiv	21	It first designates a set of surface keypoints on target objects and then trains a keypoint detector to localize them	YOLOv3 for bounding box detection; ResNet-101 as	RGB image	No	PnP algorithm

Table 1 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
					backbone of the keypoint detector			
Zhao et al. (2020a)	2020	Neurocomputing	1	It recovers the pose from 2D keypoints and 2D bounding boxes; it is robust to occluded and cluttered scenes	Feature Pyramid Network and ResNet-101 as backbone	RGB image	No	PnP algorithm
Zhao et al. (2020b)	2020	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	2	It learns 3D keypoints from pairs of images rather than explicit 3D labelling information and 3D CAD models; it can be used as an alternative when 3D CAD models or 3D annotations are not available	OK-POSE network	RGB image	No	PnP algorithm
Nath Kundu et al. (2019)	2018	European Conference on Computer Vision	7	It utilises the geometric regularity of intraclass objects for pose estimation	A fully convolutional neural network to learn pose-invariant lo-cal descriptors; a pose estimation net-work	RGB image, 3D model	No	No
Chen et al. (2019)	2019	ArXiv	5	Pose estimation of multiple texture-less shiny parts; it proposes a new approach to generate datasets and label them automatically	MaskRCNN for object detection; Stacked hourglass network for keypoint estimation	RGB image, CAD model	No	PnP algorithm with RANSAC

Table 1 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Pavlakos et al. (2017)	Instance-based: texture-less gas canister; Class--based: PASCAL3D+dataset	Instance-based: 175 RGB-D images	Two stages	Near real-time	Instance and Class-based scenarios	Calculated using a geodesic distance	Robotic manipulation
Peng et al. (2020)	LINEMOD, Occlusion LINEMOD, Truncation LINEMOD, YCB-Video datasets	NA	Two stages	Yes	Instance	2D projection metric: 99.00 (LINEMOD), 61.06 (Occlusion LINEMOD), 58.06 (Truncation LINEMOD), 47.4 (YCB-Video); ADD metric: 40.77 (Occlusion LINEMOD), 86.27 (LINEMOD), 31.48 (Truncation LINEMOD), 73.4 (YCB-Video)	General
Zhu et al. (2021)	LINEMOD, Occlusion LINEMOD	LINEMOD: 15.783 images for 13 objects; Occlusion LINEMOD: 1.214 images for 8 objects	Two stages	Yes	Instance	2D projection metric: 99.37 (LINEMOD), 63.21 (Occlusion LINEMOD); ADD metric: 91.18 (LINEMOD), 47.23 (Occlusion LINEMOD)	General
You et al. (2021)	LINEMOD, Occlusion LINEMOD	NA	Two stages	Yes	Instance	(average) 2D projection metric: 99.11 (LINEMOD); ADD metric: 91.68 (LINEMOD), 42.33 (Occlusion LINEMOD)	Robotics, AR
Zhao et al. (2018)	LINEMOD,Occlusion LINEMOD	LINEMOD: 18.273 test images for 15 objects	Two stages	No	Instance	2D reprojection error: 94.5; ADD error: 72.6 (LINEMOD)	Robotics, VR and AR
Zhao et al. (2020a)	LINEMOD, Occlusion LINEMOD	LINEMOD: 15.783 images for 15 objects; Occlusion LINEMOD: 1.214	Two stages	No	Instance	2D-pose metric: 95.8; ADD metric: 28.9 (LINEMOD)	General

Table 1 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
		images for 9 objects of LINEMOD					
Zhao et al. (2020b)	LINEMOD, Occlusion LINEMOD	LINEMOD: 15.783 images for 13 objects; Occlusion LINEMOD: 6 objects of LINEMOD	Two stages	No	Instance	ADD metric: 30.16 (LINEMOD)	General
Nath Kundu et al. (2019)	Pascal3D+, ObjectNet3D	Pascal3D+: images for 12 objects; ObjectNet3D: 90.127 images	NA	No	Instance	Median Error: 8.53° (Pascal3D+), 5.48° ObjectNet3D; Accuracy at	Real world applications
Chen et al. (2019)	Training with synthetic images generated with blender	About 20 K images with different backgrounds containing 5 or 6 objects for object detection	Two stages	No	Instance	NA	Industrial applications

In addition to general characteristics such as year of publication, journal, number of citations and highlights of the article, the table indicates whether the approach exploits a neural network for a specific task; the input of the system; the pre-processing step and the pose refinement method, if any; what dataset it uses and its size; if it belongs to one-stage or two-stage methods; its ability to run in real-time or not; whether it works at the instance or category level; the accuracy and the research field

Table 2 Template-based Methods

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pose refinement
Ulrich et al. (2012)	2011	IEEE Transactions on Pattern Analysis and Machine Intelligence	118	Hierarchical view-based approach that combines the scale-space with the similarity-based aspect graphs; robust to noise, occlusions, and clutter, invariant to contrast changes	No	3D CAD model, single camera image	Least-square adjustment with the Levenberg-Marquardt algorithm
Konishi et al. (2016)	2016	European Conference on Computer Vision	32	Perspectively Cumulated Orientation Feature, Hierarchical Pose Trees; it can manage texture-less and shiny objects, cluttered backgrounds, and partial occlusions	No	3D CAD data, monocular image	PnP algorithm
Muñoz et al. (2016b)	2016	IEEE International Conference on Robotics and Automation (ICRA)	27	Edge-based approach, RAPID-LR and RAPID-HOG algorithms	No	3D CAD models, image	No
Tjaden et al. (2017)	2017	IEEE International Conference on Computer Vision	72	Segmentation strategy based on local color histograms with extension to pose tracking	No	Monocular image	No
Cao et al. (2016)	2016	IEEE International Conference on Robotics and Automation (ICRA)	42	Color transformation and vectorization, it restructures template matching as a large-scale matrix-matrix multiplication	No	3D model, RGB image captures	No
Muñoz et al. (2016a)	2016	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	13	Cascade Forest Template: it can manage complex texture-less objects, severe self-occlusions, and specularities	No	CAD model, RGB image	No
Payet and Todorovic (2011)	2011	IEEE International Conference on Computer Vision	169	Shape-based approach, it works at level of objects categories; new mid-level feature called Bag of Boundaries	No	Image	No
Corona et al. (2018)	2018	IEEE/RSJ International Conference on	25	Neural model to pose estimation by learning to compare real	Two branches: pose		No

Table 2 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pose refinement
		Intelligent Robots and Systems (IROS)		views and rendered ones. It manages objects with rotational symmetry and exploits rendered depth maps instead of RGB views	estimation, CAD model symmetry orders computation	3D CAD model, RGB image	
Sundermeyer et al. (2020)	2020	International Journal of Computer Vision	29	Variant of the Denoising Autoencoder, trained on simulated views using Domain Randomization	Augmented Autoencoders	RGB image	No
Massa et al. (2016)	2016	IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	101	Cross-domain adaptation approach for 2D-3D exemplar detection, which can be incorporated into a CNN-based detection pipeline	CaffeNet	3D CAD model, image	No

Method	Dataset	Dataset size	Real-time	Level	Accuracy	Research field
Ulrich et al. (2012)	NA	NA	No	Instance	Position: up to 0.12% with respect to the object distance; Rotation: up to 0.35 degree	Industrial and robotic applications
Konishi et al. (2016)	Nine texture-less Objects	500 images per object	No	Instance	Relation between the success rate of correctly estimated 6D pose and false positives per image between 0.7 and 0.9	Industrial and consumer robot applications
Muñoz et al. (2016b)	Six real texture-less objects, freely available	NA	No	Instance	NA	Bin-picking and grasping problems
Tjaden et al. (2017)	ACCV dataset	NA	No	Instance	NA	Tracking applications
Cao et al. (2016)		600 template training images	No	Instance		Robotics

Table 2 (continued)

Method	Dataset	Dataset size	Real-time	Level	Accuracy	Research field
	Custom dataset of 16 indoor 3D objects, RGB images with ground truth annotation				Accurate global view alignment, it does not capture significant deformations	
Muñoz et al. (2016a)	New framework to compute the training dataset, synthetically generated and real image sequences	10–20 MB	Yes	Instance	Average rotation error below 5 degrees for an 80% of the trials	Industrial applications
Payet and Todorovic (2011)	10 object instances	250 images per category, a total of 1944 images for testing	No	Category	Euclidean distance between the centroid of the ground truth camera and the estimated one: average error of 3.1 ± 1.2 units	Automotive
Corona et al. (2018)	Real images with CAD models dataset; industrial CAD models dataset	27.458 real images of 17 different types of objects; 6.660 CAD models	No	Instance	NA	Industrial and robot applications
Sundermeyer et al. (2020)	T-LESS, LINEMOD	NA	Yes	Instance	ADD metric: 32.63 (LINEMOD)	Robotic manipulation, AR
Massa et al. (2016)	IKEA, Pascal VOC subset	NA	No	Instance	Azimuth angle within 20° of the ground truth for 90% of the examples	General

In addition to general characteristics such as year of publication, journal, number of citations and highlights of the article, the table indicates whether the approach exploits a neural network for a specific task; the input of the system; the pose refinement method, if any; what dataset it uses and its size; its ability to run in real-time or not; whether it works at the instance or category level; the accuracy and the research field

Table 3 BB prediction and PnP algorithm-based Methods

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
Rad and Lepetit (2017)	2017	IEEE International Conference on Computer Vision	402	Holistic approach which predicts the pose directly from its appearance	Cascade of multiple CNNs	RGB image	No	PnP algorithm
Oberweger et al. (2018)	2018	European Conference on Computer Vision (ECCV)	100	To manage severe occlusions this approach calculates heatmaps from small patches independently and then combines them to obtain robust predictions	Deep Network to output 2D projections	RGB image	No	PnP algorithm
Liu and He (2019)	2019	ArXiv	6	End-to-end algorithm; it introduces the Collinear Equation Layer to output the 2D projections of the 3D Bounding Box corners	Fully Convolutional Network	RGB image	No	No
Liu and He (2021)	2019	Sensors 2021	2	It introduces the Bounding Box Equation algorithm to obtain the 3D translation from 3D rotation and 2D Bounding Box	Q-Net: novel CNN to regress the unit quaternion	RGB image	No	No
Tekin et al. (2018)	2018	IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	416	NN with fully convolutional architecture for precise and efficient object detection and pose estimation without refinement	YOLO6D	RGB image	No	No
Hu et al. (2019)	2019	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	110	Segmentation-driven framework	CNN with two streams: one for segmentation and one for 2D keypoints regression	RGB image	No	RANSAC-based PnP strategy

Table 3 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
Li et al. (2019)	2019	IEEE/CVF International Conference on Computer Vision (ICCV)	79	CDPN treats rotation and translation differently; it is highly accurate and can manage texture-less and occluded objects	Coordinate-Based Disentangled Pose Network	RGB image	No	PnP algorithm
Park et al. (2019)	2019	IEEE/CVF International Conference on Computer Vision (ICCV)	99	Pix2Pose predicts the 3D coordinates of each object pixel without textured models; it introduces a novel loss function to manage symmetric objects	Auto-encoder architecture	RGB image	No	PnP algorithm with RANSAC iterations
Zakharov et al. (2019)	2019	IEEE/CVF International Conference on Computer Vision (ICCV)	112	DPOD regresses multi-class object masks and dense 2D-3D correspondences between image pixels and 3D models	Encoder-Decoder network	RGB image	No	PnP algorithm with RANSAC; CNN
Li et al. (2020)	2020	ArXiv	2	It involves Domain Randomization to manage occlusions and the lack of labelled data	Self-supervised Siamese Pose Network	RGB image	No	PnP algorithm with RANSAC
Zhang et al. (2019)	2019	Image and Vision Computing	4	It can manage texture-less and occluded objects	End-to-end deep learning architecture, Feature Pyramid on top of ResNet-h	RGB image	No	PnP algorithm
Kastner et al. (2020)	2020	17th International Conference on Ubiquitous Robots (UR)	2	It provides a more automated method of calibrating the AR devices with mobile robotic systems; it introduces an annotation tool	YOLOv3 client-server architecture	RGB image	No	PnP algorithm
Liu et al. (2020)	2019	IEEE Sensors Journal	2	Feasible and highly efficient	TQ-Net	RGB image	No	NA
Yang et al. (2021)	2021	IEEE/CVF Conference on Computer Vision	1	Weakly- and self-supervised framework, it employs a differential renderer	Dual Scale Pose Estimation Network	RGB image, 2D object annotations	2D Bounding Box detection	PnP algorithm

Table 3 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
		and Pattern Recognition (CVPR)						

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Rad and Lepetit (2017)	LINEMOD, Occlusion, T-LESS	NA	Two stages	No	Instance	2D projections: 85.3 (LINEMOD); 6D Pose: 54.0 (LINEMOD); 5 cm 5° Metric: 58.6 (LINEMOD)	Robotics, VR and AR applications
Oberweger et al. (2018)	Real and synthetic training data belonging to LINEMOD; Occluded LINEMOD, YCB-VIDEO	Occluded LINEMOD: 1.215 frames; YCB-VIDEO: 92 video sequences	Two stages	No	Instance	2D reprojection error (percentage of frames having error 5px): 60.9 (Occluded LINEMOD), 39.4 (YCB-VIDEO); ADD metric (percentage of frames having error 10% of object diameter): 30.4 (Occluded LINEMOD), 53.1 (YCB-VIDEO)	General
Liu and He (2019)	LINEMOD	NA	One stage	No	Instance	2D projection error: (average pixel) 2.71; rotational and translational error: (average) 1.66 cm, 2.43°	General
Liu and He (2021)	LINEMOD	NA	One stage	No	Instance	Average Euler angle error: (average) 5.21°; 5 cm 5° Metric: 58.07	General
Tekin et al. (2018)	LINEMOD; OCCLUSION	LINEMOD: 15.783 images for 13 objects	One stage	Yes	Instance	IoU score: 99.92; ADD metric: 55.95 (LINEMOD)	AR and VR, robotics
Hu et al. (2019)	Occluded LINEMOD, YCB-VIDEO	NA	Two stages	Yes	Instance	2D reprojection error 5px: 44.9 (Occluded LINEMOD), 30.8 (YCB-VIDEO); ADD metric 10% diameter: 27.0 (Occluded LINEMOD), 39.0 (YCB-VIDEO)	AR, robot manipulation
Li et al. (2019)	LINEMOD, Occlusion	NA	Two stages	Yes	Instance	2D projections: 98.10 (LINEMOD); ADD: 89.86(LINEMOD); 5 cm 5° Metric: 94.31 (LINEMOD)	AR, robot manipulation

Table 3 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Park et al. (2019)	LINEMOD, LINEMOD Occlusion, T-LESS	NA	Two stages	No	Instance	ADD-10%: 72.4 (LINEMOD), 32.0 (LINEMOD Occlusion); Visible Surface Discrepancy: 29.5 (T-Less)	AR,robot manipulation
Zakharov et al. (2019)	Real and synthetic training data; LINEMOD, OCCLUSION	NA	Two stages	Yes	Instance	ADD-10%: 66.43 (synthetic), 95.15 (real) (LINEMOD); 47.25 (OCCLUSION)	General
Li et al. (2020)	LINEMOD, Occluded LINEMOD, YCB, Randomization LINEMOD	LINEMOD: 15 poorly textured objects; YCB: 21 gadgets observed in 92 sequences	Two stages	No	Instance	2D projection: 90.9 (LINEMOD), 15.6 (YCB); ADD: 81.1 (LINEMOD), 51.3 (Occluded LINEMOD); ADD-AUC: 50.5 (YCB); 5 cm 5° Metric: 85.7 (LINEMOD), 24.7 (YCB)	AR, mobile robotics, autonomous navigation
Zhang et al. (2019)	LINEMOD, OCCLUSION	NA	Two stages	Yes	Instance	2D reprojection: 94.68 (LINEMOD), 45.0 (OCCLUSION); IoU: 99.66 (LINEMOD); ADD: 71.70 (LINEMOD), 24.0 (OCCLUSION); 5 cm 5° Metric: 84.38 (LINEMOD), 13.7 (OCCLUSION)	AR, autonomous driving, robotics
Kastner et al. (2020)	Custom dataset made of real and artificial images	13.069 images: 6.994 artificial, 6.075 real	Two stages	Yes	Instance	ADD metric: about 93	AR devices, robotic systems
Liu et al. (2020)	ACCV	18.000 real images	Two stages	Yes	Instance	Average pixel projection error: 2.72	AR, robot manipulation
Yang et al. (2021)	LINEMOD and Occluded LINEMOD with additional 10 K synthetic data; HomebrewedDB	NA	Two stages	No	Instance	ADD: 58.6 (LINEMOD); 24.8 (Occluded LINEMOD); 44.0 (HomebrewedDB)	General

In addition to general characteristics such as year of publication, journal, number of citations and highlights of the article, the table indicates the neural network exploited by the approach; the input of the system; the pre-processing step and the pose refinement method, if any; what dataset it uses and its size; if it belongs to one-stage or two-stage methods; its ability to run in real-time or not; whether it works at the instance or category level; the accuracy and the research field

Table 4 Classification-based Methods

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
Kehl et al. (2017)	2017	IEEE/CVF International Conference on Computer Vision (ICCV)	508	It follows the SSD paradigm	Derived from InceptionV4	RGB image	Offline stage to precompute Bounding Boxes	ICP approach
Su et al. (2015)	2015	IEEE/CVF International Conference on Computer Vision (ICCV)	666	CNN trained with synthetic 3D objects superimposed on real backgrounds	CNN	RGB image, 3D model	No	No
Josifovski et al. (2018)	2018	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	11	It combines the robustness of CNNs with a fine-resolution instance-based 3D pose estimation	R-CNN for object detection; VGG-16 architecture for viewpoint estimation	RGB image, 3D model	No	No
Li et al. (2019)	2019	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	133	3D guidance: basic cuboid, then refined	2D+O subnet for object detection: modified version of an R-CNN; 3D subnet	RGB image	No	Yes
Poirson et al. (2016)	2016	Fourth International Conference on 3D Vision (3DV)	101	It does not require resampling of the image because it detect the pose in a single forward step	SSD, VGG	RGB image	No	No
Mousavian et al. (2017)	2017	IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	564	It estimates the position and size of an object from its bounding box and surrounding pixels	CNN, VGG network	RGB image	No	No
Xu et al. (2018)	2018	IEEE Conference on Computer Vision and	158	Multi-level fusion scheme: it first generates the 2D region proposals, then	RPN, Faster R-CNN	RGB image	No	No

Table 4 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
		Pattern Recognition (CVPR)		simultaneously predicts position, orientation, and dimensions of the objects				
Zou et al. (2021)	2021	IEEE Transactions on Consumer Electronics	0	It detects the traffic participants while simultaneously regressing their 3D translation and rotation vectors	6D-VNet	RGB image	No	No

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Kehl et al. (2017)	Training with real and synthetic data; LINEMOD	NA	One stage	No	Instance	IoU: 99.4; ADD: 76.3	AR, robot manipulation
Su et al. (2015)	Synthesized images for training, real for testing; PASCAL 3D+	30 K models, 2.4 M images	One stage	NA	Instance	Average Viewpoint Precision: 39.7 (4 bins); 19.8 (24 bins)	General
Josifovski et al. (2018)	Custom dataset made of fully annotated synthetic data for training; T-LESS for testing	NA	One stage	NA	Instance	NA	Robotic tasks
Li et al. (2019)	KITTI	7.481 training and 7.518 test images	One stage	No	Instance	Average Precision: 90.02; Average Orientation Similarity: 89.13	Autonomous driving
Poirson et al. (2016)	Pascal 3D+	NA	One stage	Yes	Category	24 View AVP metric: 27.7	Navigation and robotics
Mousavian et al. (2017)	KITTI, Pascal 3D+	NA	One stage	NA	Instance	(KITTI) AOS: 88.75%; AP: 89.04%; OS: 0.0067 (moderate);	Robotic applications

Table 4 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
						(Pascal 3D+) MedErr: 11.1, $Acc_{\frac{\pi}{6}}$: 0.8103	
Xu et al. (2018)	KITTI	7.481 training, 7.518 testing	One stage	NA	Instance	NA	General
Zou et al. (2021)	Pascal3D+	NA	One stage	No	Instance	Average Viewpoint Precision (quantization of angles): 72.2 (4), 63.0 (8), 54.8 (16), 48.7 (24)	Autonomous driving

In addition to general characteristics such as year of publication, journal, number of citations and highlights of the article, the table indicates the neural network exploited by the approach; the input of the system; the pre-processing step and the pose refinement method, if any; what dataset it uses and its size; if it belongs to one-stage or two-stage methods; its ability to run in real-time or not; whether it works at the instance or category level; the accuracy and the research field

Table 5 Regression-based Methods

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
Xiang et al. (2018)	2017	Conference: Robotics: Science and Systems 2018	695	A single NN estimates semantic labels, 3D translation, and 3D rotation; it is robust to occlusion and can handle symmetric objects with the introduction of the Shape-Match-Loss function	PoseCNN	RGB image	No	No
Manehdran et al. (2017)	2017	IEEE International Conference on Computer Vision (ICCV)	143	It focuses on rotation between the object and the camera	Modified version of the VGG-M network	RGB image	No	No
Do et al. (2018)	2018	ArXiv	77	End-to-end deep learning pipeline which simultaneously detect, segment, and estimate the pose	RPN, Mask-RCN	RGB image	No	No
Hara et al. (2017)	2017	ArXiv	55	The first two approaches represent an orientation as a 2D point on a unit circle; the third converts the continuous orientation task into a set of discrete orientation estimation tasks (final model)	DCNN: ResNet-101	RGB image	No	No
Rambach et al. (2018)	2018	IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)	17	It uses pencil images for training the CNN, and introduces a new loss feature (ADD loss)	Modified PoneNet	RGB syn-thetic image	No	No
Wu et al. (2018)	2018	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	24	Segmentation network trained on real RGB images; Pose interpreter network trained entirely on synthetic pose data; it introduces a novel loss function	DRN for semantic segmentation; Pose interpreter network: ResNet-18 feature extractor followed by a multi-layer perceptron	RGB image	No	No
Ku et al. (2019)	2019	IEEE/CVF Conference on Computer Vision and	111		MS-CNN	RGB image	No	No

Table 5 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
		Pattern Recognition (CVPR)		It uses proposals to reduce the search space and leverages shape reconstruction through a point cloud				
Hu et al. (2020)	2020	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	28	It directly regresses the position from groups of 2D-3D correspondences associated with each keypoint	CNN	RGB image, 3D model	No	No
Wang et al. (2019)	2019	IEEE International Conference on Robotics and Biomimetics (ROBIO)	2	Deep learning multitasking end-to-end network consisting of two parts	Mask R-CNN semantic segmentation network; 6D object estimation network	RGB image	No	No
Liu et al. (2019)	2019	IEEE Transactions on Multimedia	2	Framework for pose estimation of texture-less objects under cluttered or occluded environments	Triplet network	RGB image	No	No
Capellen et al. (2020)	2019	15th International Conference on Computer Vision Theory and Applications	8	Architecture derived from PoseCNN; it replaces the quaternion estimation of PoseCNN with a fully convolutional architecture	ConvPoseCNN	RGB image	No	No
Wang et al. (2021)	2021	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	2	It learns the 6D pose from dense correspondence-based intermediate geometric representations	Geometry-guided Direct Regression Network	RGB image	No	No
Trabelsi et al. (2021)	2021	IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)	5	Pose Proposal Network followed by a refinement module	Pose Proposal Network, Pose Refinement Network	RGB image	No	Multi-Attentional Refiner (MARN)
Hu et al. (2021)	2021	IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	0	Single-stage hierarchical end-to-end trainable network for space poses; they introduce the SwissCube dataset	Feature Pyramid Network	RGB image	No	RANSAC-based PnP
Su et al. (2021)	2021	Sensors	2	Domain adaptation scheme which transforms real and synthetic images into an	SynPo-Net	RGB image	No	No

Table 5 (continued)

Method	Year	Journal	Citations	Highlights	Neural Network	Input	Pre-processing	Pose refinement
				intermediate domain that is better fit for establishing correspondences				

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Xiang et al. (2018)	It introduces the YCB-Video dataset; Occluded LINEMOD	YCB-Video: 133.827 images; real and synthetic images for training	One stage	NA	Instance	ADD metric: 53.7, ADD-S metric: 75.9 (YCB-Video); Occluded LINEMOD: 24.9	Robotic tasks
Manehdran et al. (2017)	Pascal 3D+	NA	One stage	NA	Instance	Median geodesic angle error: 15.38 (axis-angle), 16.63 (quaternion)	Autonomous navigation, 3D scene understanding
Do et al. (2018)	LINEMOD (single object); Multiple object instance pose dataset	NA	One stage	NA	Instance	2D-pose metric: 99.3 (single), 99.3 (multiple); 5cm5°: 68.5 (single), 64.5 (multiple); ADD metric: 65.2 (single), 62.0 (multiple)	Robotic applications
Hara et al. (2017)	EPFL Multi-view Car Dataset; TUD Multi-view Pedestrian Dataset	EPFL: 2.358 training, 1.120 testing; TUD: 4.732 training, 290 validation, 309 testing	One stage	NA	Instance	(EPFL) MeanAE: 9.86°; MedianAE: 3.14°; (TUD) MeanAE: 26.6°; Accuracy-22.5°: 70.6; Accuracy-45°: 86.1	Autonomous driving
Rambach et al. (2018)	LINEMOD	NA	One stage	NA	Instance	ADD error: 10.22 (10%), 38.22 (30%)	AR, robotics
Wu et al. (2018)	Oil Change Dataset; Synthetic Image Dataset	Oil Change Dataset: 7.879 training, 1.950 testing; Synthetic Image Dataset: 3.2 million training, 3.200 testing	One stage	Yes	Instance	Success (%): 71.01	Robotics

Table 5 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
Ku et al. (2019)	KITTI	NA	One stage	NA	Instance	(Hard) BEV AP: 15.78 (cars), 10.54 (pedestrians), 11.35 (cyclists); 3D AP: 9.06 (cars), 10.08 (pedestrians), 9.93 (cyclists)	Autonomous driving
Hu et al. (2020)	Occluded LINEMOD, YCB-Video	Occluded LINEMOD: 1.214 testing; YCB-Video: 130 K real images	One stage	No	Instance	ADD-0.1d: 43.3 (Occluded LINEMOD), 53.9 (YCB-Video); REP-5px: 62.3 (Occluded LINEMOD), 48.7 (YCB-Video)	General
Wang et al. (2019)	YCB-Video	80 K synthetically rendered images	One stage	No	Instance	ADD metric: 70.3	Robotics
Liu et al. (2019)	LINEMOD	NA	One stage	No	Instance	Matching score metric: 0.986	Multimedia processing, AR
Capellen et al. (2020)	YCB-Video	133.936 images, 80.000 synthetic images	One stage	NA	Instance	ADD: 57.4; ADD-S: 79.2	Robotics
Wang et al. (2021)	LINEMOD, Occluded LINEMOD, YCB-Video	LINEMOD: 13 sequences, each containing about 1.2 K images; Occluded LINEMOD: 1.214 images from a LM sequence; YCB-Video: 110 K real images	One stage	Yes	Instance	ADD: 62.2 (Occluded LINEMOD), 60.1 (YCB-Video); AUC of ADD: 84.4	General
Trabelsi et al. (2021)	YCB-Video, LINEMOD, LINEMOD Occlusion	YCB-Video augmented with 80 K synthetically rendered images; LINEMOD: 15.783 images	One stage	NA	Instance	2D projection error: 55.6 (YCB-Video), 99.19 (LINEMOD) 65.46 (Occlusion LINEMOD); ADD: 73.6 (YCB-Video), 93.87 (LINEMOD), 58.37 (Occlusion	General

Table 5 (continued)

Method	Dataset	Dataset size	Type	Real-time	Level	Accuracy	Research field
						LINEMOD); ADD AUC: 83.1 (YCB-Video),	
Hu et al. (2021)	SPEED dataset, Occluded LINEMOD; SwissCube dataset	SPEED: 10 K training, 2 K testing; SwissCube: 50 K images	One stage	No	Instance	$e_q + e_t$: 0.010 (SPEED); ADI-0.1d accuracy: 47.9 (SwissCube), 48.6 (Occluded LINEMOD)	Space applications
Su et al. (2021)	Trained only with synthetic images; LINEMOD, TUD-L	NA	One stage	NA	Instance	ADD: 44.13 (LINEMOD); bop performance score: 47.67 (TUD-L)	Robotic interaction, AR

In addition to general characteristics such as year of publication, journal, number of citations and highlights of the article, the table indicates the neural network exploited by the approach; the input of the system; the pre-processing step and the pose refinement method, if any; what dataset it uses and its size; if it belongs to one-stage or two-stage methods; its ability to run in real-time or not; whether it works at the instance or category level; the accuracy and the research field

– *Dataset size*. The size of the dataset is a helpful parameter because some specific strategies, such as training a neural network, require many data to perform satisfactorily.
– *Real-time*. Information about processing time is essential as a guideline for a new application since it is a fundamental prerequisite in some domains, such as autonomous driving.
– *Accuracy*. Accuracy is essential for some critical research fields, such as medicine. Consequently, the appropriate level of precision should be determined considering the specific application.

3 Feature-based methods

Methods in this category take advantage of local features (keypoints, grey values, edges, or intersections of straight lines) extracted from the regions of interest or all pixels in the image, and then compared with the features found on a 3D model of the object to establish 2D-3D matches [16, 20, 66]. Therefore, the pipeline includes two stages: the first stage extracts local features and compares them with 3D keypoints; the second stage involves 2D-3D correspondences to solve a geometric problem, e.g. via the PnP algorithm, to obtain the 6D position [69]. These techniques combine traditional CV approaches with CNNs. CNNs are harnessed in different stages of the pipeline to improve the overall performance of the system. Figure 2 presents a schematic illustration of these methods.

Advantages:

- They are fast and robust to the occlusions between objects and cluttered scenes [27, 55, 62, 69, 70].

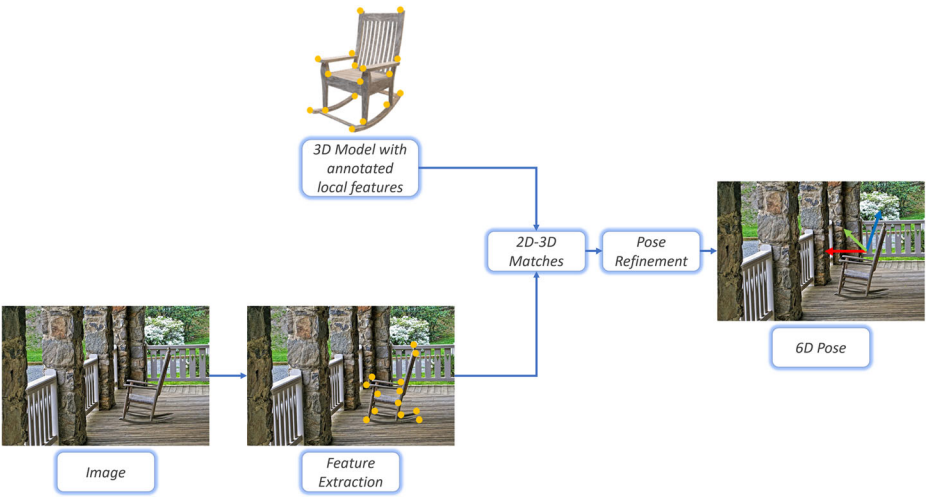

Fig. 2 Schematic representation of feature matching methods. The input image is first passed through a feature extraction step; then the extracted features are compared with those annotated on a 3D model of the object to find 2D-3D matches. Finally, 2D-3D correspondences are fed to a pose refinement step, which estimate the 6D position by solving a geometric problem

Disadvantages:

- Objects should have rich, well-defined and distinctive textures for computation of local features [25, 27, 62, 70];
- They do not work well with symmetrical objects [27];
- The quality of extracted keypoints directly affect the accuracy of position estimation [27, 65];
- Usually, these methods require a multi-state pipeline which takes much time to perform the task because 2D-3D matches generate a coarse 6D position, so they generally need a supplementary stage to obtain the final pose [7, 27].

In [41, 43], the authors used CNNs to extract the features, and a shape fitting algorithm for determining the final position. In particular, the system in [41] proposed a pipeline including object detection, keypoint location, and pose refinement. Peng et al. [43] introduced a CNN, called Pixel-wise Voting Network (PVNet), to predict the 2D-3D correspondences by regression of pixel-wise vectors to keypoints. The output was a spatial probability distribution for each keypoint, then fed to a PnP algorithm to obtain the result. This work was robust to occlusion while running at a real-time frame rate. Zhu et al. [71] tried to further improve the performance of PVNet in case of severe occlusions by introducing the Atrous Spatial Pyramid Pooling and Distance-Filtered PVNet. Furthermore, You et al. [65] built a system on top of PVNet, which used a projection loss and a discriminative refinement network to obtain a good performance.

Like most of these methods, Zhao et al. [68] leveraged a multi-stage pipeline. They identified the target object using YOLOv3, selected a set of keypoints on the target object, and then trained a ResNet101-based keypoint detector (KPD) to locate them. The 6D pose was then retrieved using a PnP algorithm fed with the 3D keypoints correspondences.

Following the pipeline categorization of the previous methods, also Zhao et al. [69, 70] refined the final output by means of geometrical algorithms. For the first part, instead, they employed a CNN to implement both object detection and keypoints estimation. In [69], the authors introduced an end-to-end framework with a ResNet architecture trained with viewpoint transformation information and salient regions. The goal was to learn geometrically and semantically consistent viewpoints. In [70], the same authors proposed OK-POSE (Object Keypoint-based pose estimation) network, which learned 3D keypoints from relative transformations between pairs of images rather than from explicit 3D labelling information and 3D CAD models.

In some cases, to remedy the lack of training data, systems trained the network with synthetic images. In this context, Nath Kundu et al. [18], introduced a two-stage pipeline: a CNN learned the local descriptors position invariant to obtain the corresponding keypoints; a second CNN, by joining the information coming from multiple correspondence maps, provided in the output the final pose estimation.

Finally, concerning the elaboration of object which are usually complex to treat, the system proposed by Chen et al. [4], focused on metallic targets, texture-less and with shiny materials. The process included three stages: object detection, feature detection, and pose estimation.

Table 1 shows a schematic recap of each analyzed work according to the main features described in the Section 2.

4 Template-based methods

These methods include a first off-line stage which build a template database from a 3D model of the object. This database includes a set of synthetic renderings, obtained by varying position and orientation, resulting in a group of patches from different points of view. These patches could be imagined as distributed over a virtual sphere surrounding the 3D model of the object. The second stage is a test phase, executed on-line to establish the 6D position. So, the current image is compared with all the database patches generated in the previous step through a sliding window algorithm. These systems use a similarity value to compute the best match, chosen by the method itself [48, 66, 69, 73]. Figure 3 shows an example of how these methods operate.

Advantages:

- They work well in case of texture-less objects [27, 69, 71];
- If the database is exhaustive, they can achieve high accuracy [6].

Disadvantages:

- They are very sensitive to variations in lighting and occlusions between items, as these circumstances affect the rate of similarity, which is very low when the lighting is scarce or when the object is occluded [7, 25, 27, 69, 71];
- The execution speed is inversely proportional to the number of elements belonging to the template [28]. However, this number is directly proportional to the accuracy of the method [35, 36]. A rich set of images is required to cover as many positions of the object as possible and to have a high probability of obtaining the correct pose. Therefore, a trade-off

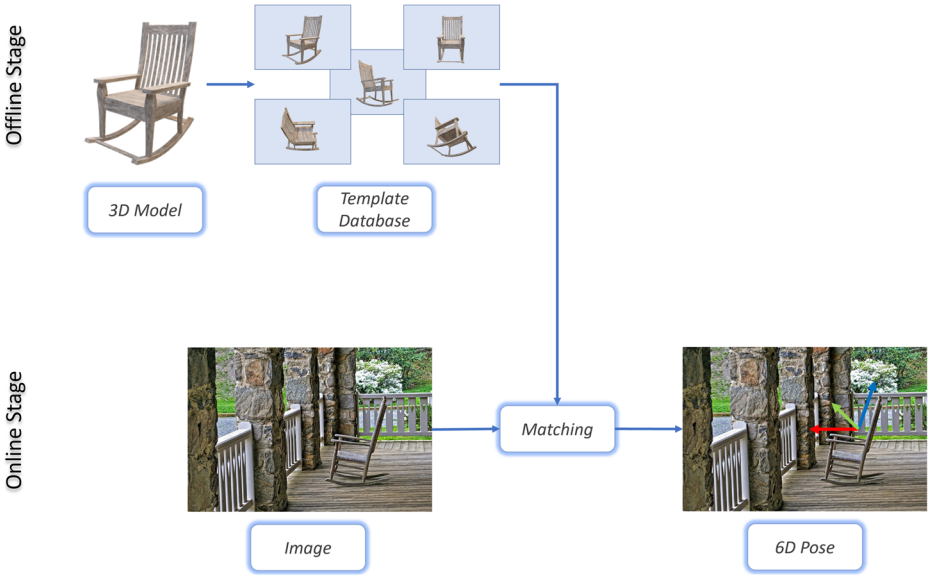

Fig. 3 Schematic representation of template matching methods. A first offline phase builds a template database from a 3D model of the object; then, during an online phase, the input image is compared with all the elements in the template to calculate the 6D position as the best match

between performance degradation and desired accuracy is required [6]. Many approaches, based on CNNs, implement changes to the cost functions by adding ad-hoc terms to solve these problems [28].

As already discussed, older methods applied geometric approaches. In this context, in [1], the authors employed color transformation and vectorization for a more compact representation, and the best match calculation. Unlike most methods that try to recover the position of a known instance, called instance-based methods, in [42], the authors worked at the level of object categories trying to estimate the position of unknown instances. They introduced features called Bags of Boundaries (BOB), which looked for matching only on a summary of edges. Edges were also used by Ulrich et al. [58], where they estimated first the discrete position, which was then refined using a 2D match, based on edge features, and the corresponding 3D camera position, using the Levenberg-Marquardt algorithm (LMA) [33]. The system automatically produced a hierarchical model from the 3D CAD model of the object to find the item in the image in an efficient and time-saving way. Another solution to reduce the execution time problem, was introduced by Konishi et al. [16], named the Perspectively Cumulated Orientation Feature (PCOF), which was used to handle a specific range of 3D object positions. Moreover, Hierarchical Pose Trees (HPT) were constructed by clustering the 3D object poses and reducing templates resolution.

A big obstacle of this class of approaches is the handling of texture-less objects. To overcome this, Muñoz et al. [36] used edge matches, given coarse position information by a detector. In [35], instead, the authors involved the Cascade Forest Template (CFT). They used regression forests for each template to learn the misalignment between the initial layout and the current one. 6D position estimation strategies are basic also for tracking. In this context, pose estimation is used for tracking initialization and pose recovery when the algorithm loses the object due to occlusions or when it comes out of the camera's point of view. For this purpose, in [56], a new segmentation strategy was proposed, based on a consistent local color histogram.

Another problem of applications requiring the exact position of an object is the management of symmetrical objects. Corona et al. [5] addressed this problem by introducing a particular loss function. As mentioned earlier, the most recent methods use neural networks to make the algorithms adopted more efficient and performing. In this case, a CNN received an RGB image and a depth map for each viewpoint, corresponding to the model renderings, to predict the 6D position. In [51], the authors used a particular neural network called denoising autoencoder for 6D pose estimation, trying to learn representations from rendered 3D model views. Finally, neural networks were used also in [31], where the authors proposed a cross-domain adaptation approach, which trained the same CNN, CaffeNet, for both the off-line and the on-line stages.

An overview of the described methods is represented in Table 2.

5 Direct prediction or learning-based methods

These methods predict the 6D pose using CNNs [7, 69], hence needing a training phase which requires large amounts of labelled data but allows CNNs to produce significant improvements for the 3D position and rotation estimation. DL based methods can be one-stage and two-stage, depending on the use of a further step to refine pose parameters through Perspective-n-Point (PnP) algorithm (Fig. 4) [73]. PnP algorithm could provide worse results when correspondences are degenerated because of occlusions [27]. In general, two-stage CNNs are more

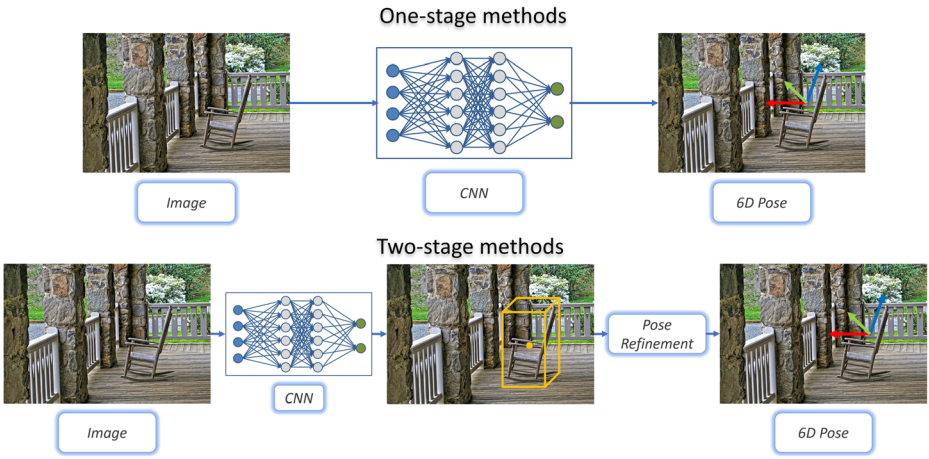

Fig. 4 One-stage and two-stage methods. One-stage methods directly output the 6D Pose from the input image by using a CNN; Two-stage methods require a further step, fed with the CNN output, to refine pose parameters

accurate than single-shot ones, notably on small objects and multiple objects. Computing the cell size and the number of items occupying the same cell is challenging in single-shot object detectors. Moreover, with many objects, occlusions between them affect the precision of some single-shot methods, which employ correspondences between an object's 3D bounding box corner and its 2D projection [27].

Specific strategies try to solve the lack of training data problem by involving synthetic images for training or processes such as data augmentation, known as Domain Randomization in the context of 6DoF (Degrees of Freedom) pose estimation. For example, as described in [23], this requires complement data with semi-realistic synthetic images. To do so, the authors rendered a 3D model of the object on a real background and then applied different augmentation techniques, such as varying lighting conditions, contrast, blur, and occlusion by

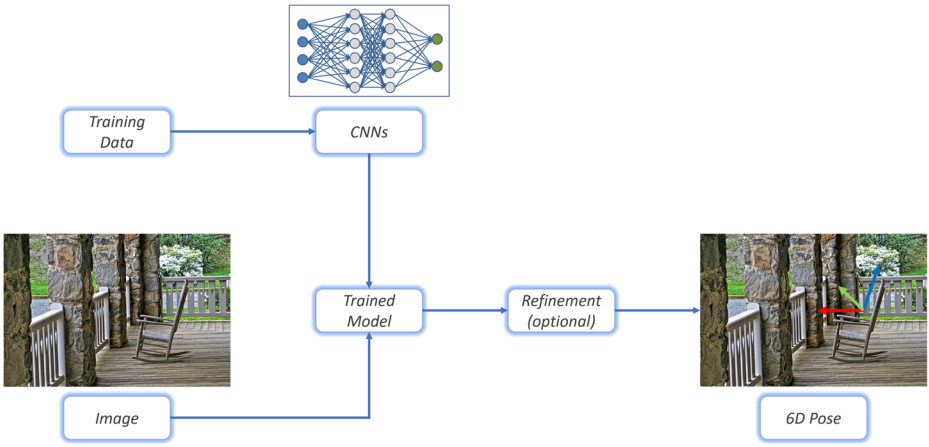

Fig. 5 Schematic representation of learning-based methods. In a first training phase, the CNN is trained with a large set of labelled data; then the trained model from an input image can estimate the 6D position, optionally refined through a refinement step

removing small image blocks and replacing them with monochrome patches. While Domain Randomization improves the pose estimation accuracy, its benefits on real test images remain limited, mostly because existing Domain Randomization strategies do not tackle the severe occlusions problem, which is one of the main challenges in pose estimation. Figure 5 shows a schematic overview of these techniques.

Learning-based methods can be classified into three categories [69, 70]:

- Bounding box prediction and PnP algorithm-based methods (Section 5.1).
- Classification-based methods (Section 5.2).
- Regression-based methods (Section 5.3).

Advantages:

- They are powerful and can provide excellent results [46, 66];
- They have high performance even if the object is partially occluded or in case of cluttered backgrounds [6].

Disadvantages:

- They require a time-consuming training process [6, 12];
- They are not very robust to severe occlusions because covering the space of all possible occlusions with real images is unmanageable [23, 39];
- Their ability to generalize is still a problem in some cases [43, 51, 73].

5.1 Bounding box prediction and PnP algorithm-based methods

These approaches use a pipeline for the 6D pose prediction composed of a CNN architecture for the object category detection and the object projected bounding box vertices prediction [69]. The methods belonging to this category are two-stages, i.e. in a first stage, they regress the projection of the corresponding 3D keypoints of the target object in the 2D image and then, in the second stage, calculate the actual 6D pose using the PnP algorithm [73]. All the systems described have the last stage in common, so they differ only on how they prepare data, i.e., the 2D-3D correspondences fed as input to the PnP algorithm for obtaining the final estimate. These methods require expensive manual annotations on bounding boxes [69].

Rad and Lepetit [45] proposed BB8, a cascade of multiple CNNs for object pose estimation task. A first CNN performed semantic segmentation, a second CNN predicted the eight corners of the 3D Bounding Box projections and, finally, after PnP algorithm processing, a third CNN per object refined the pose. The authors of BB8 extended their work in [37], trying to manage position estimation in case of severe occlusions. The proposed solution calculated heatmaps from small patches independently and then combined them to obtain robust predictions. Liu and He [24, 25] exploited the advantages of BB8 for regression. They tried to avoid the use of PnP algorithm for reducing errors and implementation consumption. The former introduced a novel layer, called the Collinear Equation layer, which provided a 2D projection of 3D bounding box angles and a new representation of the 3D rotation. The latter exploited a new algorithm, called Bounding Box Equation, to achieve accurate and efficient translation. Tekin et al. [55] based on YOLO [47] and BB8 ideas, introduced YOLO6D, a neural network with fully convolutional architecture capable

of efficient and precise object detection and pose estimation without refinement. As for BB8, the key feature here was to perform the regression of reprojected bounding box corners in the image. Moreover, in contrast to SSD6D [14], it did not suffer from pose discretization resulting in much more accurate pose estimates without refinement.

Most techniques are potentially vulnerable to occlusion as they treat the object as a global entity and calculate a unique pose estimate. In contrast, Hu et al. [9] introduced a segmentation based 6D pose estimation framework in which each visible part of objects contributed to the 2D keypoints estimation performed by a local pose predictor. To obtain a more robust and accurate result, Li et al. [21] introduced CDPN. This method treated separately the prediction of rotation and translation. It handled occluded or texture-less objects and resolved rotation using a two-stage object-level coordinate estimation and a Masked Coordinate-Confidence Loss (MCC loss). Translation was estimated directly from the image using a Scale-Invariant Translation Estimation (SITE). The approach was very accurate, fast, and scalable. Another solution, named Pix2Pose, was introduced by Park et al. [39] and predicted the 3D coordinates of each object pixel using 3D models without textures during training. The method estimated 3D coordinates and errors per pixel using an auto-encoder architecture. These pixel-wise predictions were then used in multiple stages to calculate 2D-3D matches and obtain the final pose. The method introduced a novel loss function called transformer loss for managing occlusions and symmetries. Also Zakharov et al. [66] proposed a new system called DPOD: Dense Pose Object Detector. An encoder-decoder network regressed the mask and 2D-3D matches. The training phase worked with both real and synthetic data. A finishing step, implemented via a CNN, from a coarse proposal predicted the refined one.

To manage the problem of occlusions and the lack of labelled real images, Li et al. [23] introduced a robust 6-DoF position estimation approach which exploit a Domain Randomization (DR) strategy. The method employed a first network to locate the pixels of the object. Next, a Self-supervised Siamese Pose Network (SSPN) output the coordinates and segmentation information.

The methods proposed in [13, 67], and [29] have the advantages to work in real-time. The first was an end-to-end framework, it used CNNs to obtain 2D-3D matches and worked both with texture-less objects and in case of occlusions between objects. The second exploited a client-server architecture for robots, which used YOLOv3 for object detection, keypoint detector and pose estimation. In the third, Liu et al. introduced a new network called TQ-Net. An object detection algorithm located the target and its bounding box. This information was fed into the TQ-Net to predict the translation vector T and the quaternion Q. In the end, Q was converted into a rotation matrix R. TQ-Net was easily implemented, run in real-time efficiently and accurately and worked with all previous CNN-based object detection methods.

Finally, among the most recent methods, Yang et al. [64] introduced DSC-PoseNet, to attain pose from 2D bounding-boxes. The framework learned to segment objects from real and synthetic data, then it predicted object poses through a differential renderer.

Table 3 illustrates a summary of the systems described above.

5.2 Classification-based methods

These methods aim to solve the 6D position estimation as a single-shot classification problem by discretizing the pose space. They leverage on CNNs to obtain a probability distribution in the pose space and associate it with the 3D model information to acquire the 3D position and rotation [69].

In SSD-6D [14], the authors extended the SSD detection framework [26] to 3D detection and 3D rotation estimation. A neural network performed object recognition from an RGB image returning its 2D bounding box. Each box is provided with a set of the most likely 6D poses for that instance. It decomposed a 3D rotation space into discrete viewpoints and in-plane rotations, so the rotation estimation is treated as a classification problem. In this work the authors utilized both a real dataset, for the bounding box prediction and a synthetic one for the rotation estimation. To the contrary, in [49] and [12], the authors leveraged only on a synthetic dataset. Su et al. [49] introduced a neural network trained by rendering synthetic 3D objects superimposed on real images. The trained neural network could then estimate the viewpoints of items in real situations. The method proposed in [12] combined the robustness of CNNs with high-resolution instance-based 3D pose estimation. The model used a modular architecture consisting of a detector and viewpoint estimator. The output of the architecture did not directly provide a 6DoF pose. The PnP algorithm was used to combine the intrinsic parameters of the camera and the 3D model. In opposition, in [22], the authors introduced GS3D and showed that 6D estimation could be solved without the use of a synthetic dataset. A modified Faster R-CNN detector, based on a CNN called 2D + O, classified the rotation from RGB images and the 2D Bounding Box parameters. Then, the 2D bounding box and the orientation obtained were used together with a knowledge of the guidance scenario to generate a basic cuboid called guidance, then projected onto the image plane. Another CNN called 3D Subnet received these features to refine the guidance. In the same research field, Zou et al. [72], proposed 6D-VNet to estimate traffic participants' poses for autonomous driving applications.

Most approaches we discussed separate the object detection phase from the pose estimation one, by making them run on two separate networks. These methods require resampling the image at least three times: (1) to find region proposals, (2) for detection and (3) for pose estimation. The method proposed by Poirson et al. [44] did not require resampling of the image and used convolutions to detect the object and its position in a single forward step. It provided acceleration in execution time because it did not require image resampling, and the computation for detection and pose estimation was shared. The scheme employed a Single Shot Detector. Mousavian et al. [34] estimated the position and size of an object 3D bounding box from its 2D bounding box and surrounding pixels. This method used a detector extended to regress the orientation and size of the item by training a CNN. These predictions were combined with geometric constraints to produce the final 3D pose, estimating the translation and 3D bounding box. At last, the network used by Xu et al. [63] contained two parts: one for the generation of the 2D region proposal through a Region Proposal Network (RPN), and the other for the simultaneous prediction of position, orientation, dimensions of 2D objects and 3D poses.

An overview of Classification-based approaches is shown in Table 4.

5.3 Regression-based methods

These systems solve 6D pose as a regression problem and use CNNs to estimate the position [69]. They directly regress the 6D pose parameters of the target object from the input image [73]. Usually, there is a preliminary stage of object detection to simplify the position estimation process [40]. These systems belong to the category of one-stage methods, i.e. they design a neural network which receives an input image for training and solves the posed problem by learning the rotation and 3D translation of the object represented in it [73]. PoseCNN [62] is one of the current top performers for this task in RGB images. Xiang et al. designed a fully CNN composed of two stages to jointly segment objects, estimate the

rotation, and the distance from the camera. The first two stages extracted and integrate feature maps with different resolutions from the input image. The network output semantic labels, 3D translation, and 3D rotation. PoseCNN did not address input images containing multiple instances of the same object and might require further refinement steps to improve the accuracy. The method proposed in [30] focused only on rotation between the object and the camera using a modified version of the VGG-M network. The pipeline contained a featured network and a pose network.

The methods described below, although less important than the previous ones, has their own relevance in research. Do et al. [7] introduced Deep-6DPose to detect, which was an end-to-end deep learning pipeline consisting of a RPN to derive the Regions of Interest and a Mask-RCN. It decoupled pose parameters into translation and rotation. For autonomous driving applications, Hara et al. [8] considered objects seen approximately sideways in the center of an image. The authors proposed three approaches for estimating the rotation: the first two differed only in the loss function and represented angles as points on a unitary circle and trained a regression function. The third approach employed the discretization process. For the same research field, Ku et al. [17] introduced MonoPSR, a method for 3D Object Detection that used suggestions and shapes reconstruction. Rambach et al. [46] trained a CNN to directly regress the object 6D pose using only single-channel synthetic images with improved edges, obtained from rendering the 3D object. It used a modified version of the PoseNet architecture [15] with a new loss feature to facilitate the training process. In contrast to other CNN-based approaches for pose estimation, which require many data to be trained, in [61] training was done only with synthetic position data and then extended to real data. The process consisted of two cascading components: a segmentation network (DRN: Dilated Residual Network) that generated the segmentation masks and a pose interpreter network. The image and the segmentation result were the inputs of the pose interpreter network.

In this last part, the most recent and fewer known methods are described. Hu et al. [10] assumed the objects were rigid and their 3D model was available. The proposed network directly regressed the position from groups of 2D-3D correspondences associated with each keypoint. The system used three main modules to infer the pose: a local feature extractor, a feature aggregation module, and a global inference module. The CNN proposed in [59] computed both the mask and the 6D pose. The system was divided into two distinct networks to overcome the effects caused by the lack of training data: segmentation network and pose estimation network. Liu et al. [28], used rendered binary images in the training phase to generate triplets. The triplets were fed to a triplet network to capture the features, while the positions were reference information. The regression network provided the final pose. Capellen et al. [2] introduced ConvPoseCNN, an architecture derived from PoseCNN [62], described above. At first, a VGG16 convolutional backbone extracted the features. The system performed first pixel-wise semantic segmentation through a fully convolutional branch. Then a fully convolutional vertex branch estimated central direction and depth. The results of these two branches found the center of the objects and their bounding boxes. A fully convolutional architecture, like the other two branches, replaced the PoseCNN quaternion estimation branch to estimates quaternions for each pixel. Wang et al. [60] proposed the Geometry Guided Direct Regression Network (GDR-Net) to unify direct and geometry-based indirect methods. The system first detected all the objects, and, for each detection, it zoomed in to the corresponding Region of Interest (RoI). Each RoI was fed to the network to predict several intermediate geometric feature maps. Then the Patch-PnP algorithm directly regressed the 6D object pose from Dense Correspondences and Surface Region Attention. Trabelsi et al. [57] introduced an

Table 6 The table provides a summary of the results that emerged from this work, indicating for each feature the most appropriate method among Feature-based, Template-based, and Learning-based

	Occlusions and cluttered scenes	Not well-defined texture	Symmetrical objects	Real-time	Non-exhaustive dataset	Bad lighting conditions	High accuracy	High computational power availability
Feature-based methods	X				X			
Template-based methods		X	X					
Learning-based methods	X	X	X	X		X	X	X

end-to-end 6D object pose estimation method, made by a pose proposal module and a pose refinement module. The former output an object classification and an initial pose estimation. The latter embodied a differentiable render and an iterative refiner called MARN. Hu et al. [11] involved a Feature Pyramid Network for multiple scales 6D pose regression of space objects. Finally, Su et al. [50] introduced SynPo-Net, a CNN trained exclusively with synthetic images, which tried to improve accuracy in pose estimation by replacing pooling layers with convolutional layers.

Regression-based techniques are summarized in Table 5.

6 Discussion and conclusion

6D pose estimation of an object from a single RGB image is a central issue in the computer vision community, especially after the introduction of deep learning solutions, which speeded up the diffusion of new applications. This review analyzed the most recent and relevant methods available in the literature and classified them according to the procedure adopted, to define a series of guidelines related to this problem. To summarize the methods, Tables 1, 2, 3, 4 and 5 show the main feature values of every study. The tables contain some general features of each article, such as the publication year, the journal or conference, the number of citations, the highlights, and the research field. Furthermore, the system input, the neural network, pre-processing, and refinement methods, if used, have been specified. Finally, the tables indicate the dataset and its size, the accuracy, if the method is one-stage or two-stage and whether it works at the instance or category level. These parameters, as specified in the Section 2, have been chosen as they highlight the main features of each group of methods and the differences among the different classes. For this reason, they can be used as guidelines to choose the correct approach for a new specific task, relying on previous works.

Feature-based methods (Table 1) could be the appropriate solution if the target object has a recognizable shape, but the keypoints must be accurately chosen, and a refinement step is often required [4, 41, 43, 65, 68–71]. Even though the refinement step of two-stage methods is time-consuming, some of them can run in real-time [43, 65, 71] or near real-time [41]. These systems have been evaluated on LINEMOD dataset [43, 65, 68–71], PASCAL3D + dataset [18, 41], KITTI dataset [3] or custom datasets [4].

Template-based methods (Table 2) can reach high accuracy values with an exhaustive template database [35, 58], but the matching process is time-consuming, so they rarely work in real-time [35, 51]. Except one study [42], which is also the only category-level method, all the approaches belonging to this category require a 3D model of the object in addition to the image as input. Moreover, most systems create a custom dataset and do not require pose refinement. To the contrary, in [16, 58], the methodology need a further step of refinement to estimate the pose. Finally, only three systems [5, 31, 51] involve CNNs to solve the problem.

In recent years, the research focused on Learning-based methods, which allow the training of a classification-based or a regression-based neural network tailored for a specific task. For Learning-based methods three different tables have been created, one for each subcategory described in previous sections. Apart from the methods in [24, 25], Bounding box prediction and PnP algorithm-based methods (Table 3) are two-stages, and calculate the actual 6D refined using PnP algorithm. Just in [64], the system needs a supplementary pre-processing step, which detects the 2D bounding box. Despite the time-consuming multi-stage pipeline, some approaches can work in real-time [9, 13, 21, 29, 66, 67]. Most of the studies have been

evaluated on LINEMOD dataset and its variation, named Occluded LINEMOD; some of them, instead, exploit T-LESS Dataset [39, 45], YCB-Video Dataset [9, 37], ACCV Dataset [29], or a custom dataset [64].

Classification-based methods (Table 4) are one-stage, and they rarely need a pre-processing [14] or a pose refinement step [14, 22]. The work described in [44] is the only category-based approach, and it can work in real-time. These systems have been mainly tested on PASCAL3D + Dataset [34, 44, 49, 72] and KITTI Dataset [22, 34, 63].

The most known and performant systems belong to the Regression-based methods (Table 5), which do not require either pre-processing or post-processing steps, as they directly regress the 6D position through a single-stage pipeline.

The training process of Learning-based methods is time-consuming and requires computational power. For this reason, some Regression-based systems propose end-to-end trainable networks to simplify the process and obtain real-time working methods [55, 60, 61].

Starting from these values, the most remarkable methods were analyzed and classified, deriving their main characteristics, strengths, and weaknesses. Therefore, it has been possible to put together all the pros found in the different articles and define what should be the correct approach for the 6D position estimation of an object from a single RGB image, which can work even under boundary conditions, namely, auto-occlusions, symmetries, occlusions between multiple objects, and bad lighting conditions. To summarize the findings of this work, Table 6, based on the technical prerequisites and boundary conditions for a potential future application, provides guidelines on which category among those described in Sections 3, 4 and 5 is most appropriate to meet the required needs. It could be inferred that algorithms belonging to Feature-based and Template-based methods are outdated and they could only be used as support for Learning algorithms, as single steps of a larger system based on neural networks. They could be exploited only whether a large dataset is not available, or the computational power is not enough to train a neural network. In these cases, the solution is turning to classical geometric algorithms. On the contrary, learning algorithms are getting better and better results, often employing artificial datasets, and avoiding the expensive data retrieval phase. In terms of real-time speed, accuracy, pipeline complexity, Regression-based approaches are the most performing, but, at the same time, more specific. On the other hand, the other two groups can provide a more generic scheme of implementation. Furthermore, the network's ability to generalize is still a challenge in some cases. This limit leads the research to move towards new efficient training databases and new techniques for automatic labelling to obtain increasingly accurate solutions.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Cao Z, Sheikh Y, Banerjee NK (2016) Real-time scalable 6DOF pose estimation for textureless objects. 2016 IEEE Int. Conf. Robot. Autom. ICRA, pp 2441–2448. https://doi.org/10.1109/ICRA.2016.7487396
2. Capellen C, Schwarz M, Behnke S (2020) ConvPoseCNN: dense convolutional 6D Object Pose Estimation. 15th Int. Jt. Conf. Comput. Vis. Imaging Comput. Graph. Theory Appl., vol 5, Valletta, Malta: pp 162–72. https://doi.org/10.5220/0008990901620172
3. Chen X, Kundu K, Zhang Z, Ma H, Fidler S, Urtasun R (2016) Monocular 3D object detection for autonomous driving. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, pp 2147–2156. https://doi.org/10.1109/CVPR.2016.236
4. Chen C, Jiang X, Zhou W, Liu Y-H (2019) Pose estimation for texture-less shiny objects in a single RGB image using synthetic training data. ArXiv190910270 Cs
5. Corona E, Kundu K, Fidler S (2018) Pose estimation for objects with rotational symmetry. 2018 IEEERSJ Int. Conf. Intell Robots Syst IROS, pp 7215–7222. https://doi.org/10.1109/IROS.2018.8594282
6. Dabbour AA, Habib R, Saii M (2020) Object pose estimation in monocular image using modified FDCM. Comput Sci T. 21(1). https://doi.org/10.7494/csci.2020.21.1.3426
7. Do T-T, Cai M, Pham TT, Reid I (2018) Deep-6DPose: recovering 6D object pose from a single RGB image. ArXiv
8. Hara K, Vemulapalli R, Chellappa R (2017) Designing deep convolutional neural networks for continuous object orientation estimation. ArXiv170201499 Cs
9. Hu Y, Hugonot J, Fua P, Salzmann M (2019) Segmentation-driven 6D object pose estimation. 2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp 3380–3389. https://doi.org/10.1109/CVPR.2019.00350
10. Hu Y, Fua P, Wang W, Salzmann M (2020) Single-stage 6D object pose estimation. 2020 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp 2927–2936. https://doi.org/10.1109/CVPR42600.2020.00300
11. Hu Y, Speierer S, Jakob W, Fua P, Salzmann M (2021) Wide-depth-range 6D object pose estimation in space. Proc. IEEECVF Conf. Comput. Vis. Pattern Recognit., pp 15870–15879
12. Josifovski J, Kerzel M, Pregizer C, Posniak LWermter S (2018) Object detection and pose estimation based on convolutional neural networks trained with synthetic data. 2018 IEEERSJ Int. Conf10.1109/IROS.2018.8594379Josifovski J, Kerzel M, Pregizer C, Posniak L, Wermter S (2018) Object detection and pose estimation based on convolutional neural networks trained with synthetic data. 2018 IEEERSJ Int. Conf. Intell. Robots Syst. IROS, pp 6269–6276. https://doi.org/10.1109/IROS.2018.8594379
13. Kästner L, Dimitrov DLambrecht J (2020) A Markerless Deep Learning-based 6 Degrees of Freedom Pose Estimation for Mobile Robots using RGB Data. 2020 17th Int10.1109/UR49135.2020.9144789Kästner L, Dimitrov D, Lambrecht J (2020) A markerless deep learning-based 6 degrees of freedom pose estimation for mobile robots using RGB data. 2020 17th Int. Conf. Ubiquitous Robots UR, pp 391–396. https://doi.org/10.1109/UR49135.2020.9144789
14. Kehl W, Manhardt F, Tombari F, Ilic S, Navab N (2017) SSD-6D: making RGB-Based 3D detection and 6D pose estimation great again. 2017 IEEE Int. Conf. Comput. Vis. ICCV, pp 1530–1538. https://doi.org/10.1109/ICCV.2017.169
15. Kendall A, Grimes M, Cipolla R (2015) PoseNet: a convolutional network for real-time 6-DOF camera relocalization. 2015 IEEE Int. Conf. Comput. Vis. ICCV, pp 2938–2946. https://doi.org/10.1109/ICCV.2015.336
16. Konishi Y, Hanzawa Y, Kawade M, Hashimoto M (2016) Fast 6D Pose Estimation from a Monocular Image Using Hierarchical Pose Trees. In: Leibe B, Matas J, Sebe N, Welling M (eds) Comput. Vis. – ECCV 2016. Springer International Publishing, Cham, pp 398–413. https://doi.org/10.1007/978-3-319-46448-0_24.
17. Ku J, Pon AD, Waslander SL (2019) Monocular 3D object detection leveraging accurate proposals and shape reconstruction. 2019 IEEECVF Conf. Comput Vis Pattern Recognit CVPR, pp 11859–11868. https://doi.org/10.1109/CVPR.2019.01214
18. Kundu JN, Rahul MV, Ganeshan A, Babu RV (2019) Object pose estimation from monocular image using multi-view keypoint correspondence. In: Leal-Taixé L, Roth S (eds) Comput. Vis. – ECCV 2018

Workshop. Springer International Publishing, Cham, pp 298–313. https://doi.org/10.1007/978-3-030-11015-4_23.
19. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
20. Li X, Cai Y, Wang S, Lu T (2019) Learning category-level implicit 3D rotation representations for 6D pose estimation from RGB images. 2019 IEEE Int. Conf. Robot Biomim ROBIO, pp 2310–2315. https://doi.org/10.1109/ROBIO49542.2019.8961408
21. Li Z, Wang G, Ji X (2019) CDPN: coordinates-based disentangled pose network for real-time RGB-Based 6-DoF object pose estimation. 2019 IEEECVF Int. Conf. Comput Vis ICCV, pp 7677–7686. https://doi.org/10.1109/ICCV.2019.00777
22. Li B, Ouyang W, Sheng L, Zeng X, Wang X (2019) GS3D: an efficient 3D object detection framework for autonomous driving. 2019 IEEECVF Conf. Comput Vis Pattern Recognit CVPR, pp 1019–1028. https://doi.org/10.1109/CVPR.2019.00111
23. Li Z, Hu Y, Salzmann M, Ji X (2020) Robust RGB-based 6-DoF pose estimation without real pose annotations. ArXiv200808391 Cs
24. Liu J, He S (2019) 6D object pose estimation based on 2D bounding box. ArXiv190109366 Cs
25. Liu J, He S (2019) 6D Object Pose Estimation without PnP. ArXiv190201728 Cs
26. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y et al (2016) SSD: single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M et al (eds) Comput. Vis. – ECCV 2016. Springer International Publishing, Cham, pp 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
27. Liu F, Fang P, Yao Z, Fan R, Pan Z, Sheng W et al (2019) Recovering 6D object pose from RGB indoor image based on two-stage detection network with multi-task loss. Neurocomputing 337:15–23. https://doi.org/10.1016/j.neucom.2018.12.061
28. Liu Y, Zhou L, Zong H, Gong X, Wu Q, Liang Q et al (2019) Regression-based three-dimensional pose estimation for texture-less objects. IEEE Trans Multimed 21:2776–2789. https://doi.org/10.1109/TMM.2019.2913321
29. Liu J, He S, Tao Y, Liu D (2020) Realtime RGB-based 3D object pose detection using convolutional neural networks. IEEE Sens J 20:11812–11819. https://doi.org/10.1109/JSEN.2019.2946279
30. Mahendran S, Ali H, Vidal R (2017) 3D pose regression using convolutional neural networks. 2017 IEEE Conf. Comput. Vis. Pattern Recognit. Workshop CVPRW, pp 494–495. https://doi.org/10.1109/CVPRW.2017.73
31. Massa F, Russell BC, Aubry M (2016) Deep exemplar 2D-3D detection by adapting from real to rendered views. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, pp 6024–6033. https://doi.org/10.1109/CVPR.2016.648
32. Moher D, Liberati A, Tetzlaff J, Altman DG (2010) Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. Int J Surg 8:336–341. https://doi.org/10.1016/j.ijsu.2010.02.007
33. Moré JJ (1978) The Levenberg-Marquardt algorithm: implementation and theory. In: Watson GA (ed) Numer. Anal. Springer, Berlin, pp 105–116. https://doi.org/10.1007/BFb0067700
34. Mousavian A, Anguelov D, Flynn J, Košecká J (2017) 3D bounding box estimation using deep learning and geometry. 2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, pp 5632–5640. https://doi.org/10.1109/CVPR.2017.597
35. Muñoz E, Konishi Y, Beltran C, Murino V, Del Bue A (2016) Fast 6D pose from a single RGB image using Cascaded Forests Templates. 2016 IEEERSJ Int. Conf. Intell. Robots Syst. IROS, pp 4062–4069. https://doi.org/10.1109/IROS.2016.7759598
36. Muñoz E, Konishi Y, Murino V, Del Bue A (2016) Fast 6D pose estimation for texture-less objects from a single RGB image. 2016 IEEE Int. Conf. Robot. Autom. ICRA, pp 5623–30. https://doi.org/10.1109/ICRA.2016.7487781
37. Oberweger M, Rad M, Lepetit V (2018) Making deep heatmaps robust to partial occlusions for 3D object pose estimation. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Comput. Vis. – ECCV 2018. Springer International Publishing, Cham, pp 125–141. https://doi.org/10.1007/978-3-030-01267-0_8
38. Olivetti EC, Ferretti J, Cirrincione G, Nonis F, Tornincasa S, Marcolin F, Deep (2020) CNN for 3D Face Recognition. In: Rizzi C, Andrisano AO, Leali F, Gherardini F, Pini F, Vergnano A (eds) Des. Tools Methods Ind. Eng. Springer International Publishing, Cham, pp 665–674. https://doi.org/10.1007/978-3-030-31154-4_56
39. Park K, Patten T, Vincze M (2019) Pix2Pose: pixel-wise coordinate regression of objects for 6D pose estimation. 2019 IEEECVF Int. Conf. Comput. Vis. ICCV, pp 7667–7676. https://doi.org/10.1109/ICCV.2019.00776
40. Patil AV, Rabha P (2019) A survey on joint object detection and pose estimation using monocular vision. MATEC Web Conf 277:02029. https://doi.org/10.1051/matecconf/201927702029
41. Pavlakos G, Zhou X, Chan A, Derpanis KGDaniilidis K (2017) 6-DoF object pose from semantic keypoints10.1109/ICRA.2017.7989233Pavlakos G, Zhou X, Chan A, Derpanis KG, Daniilidis K (2017)

6-DoF object pose from semantic keypoints. 2017 IEEE Int. Conf. Robot. Autom. ICRA, 2017, pp 2011–2018. https://doi.org/10.1109/ICRA.2017.7989233
42. Payet N, Todorovic S (2011) From contours to 3D object detection and pose estimation. 2011 Int. Conf. Comput. Vis., pp 983–990. https://doi.org/10.1109/ICCV.2011.6126342
43. Peng S, Liu Y, Huang Q, Zhou X, Bao H (2019) PVNet: pixel-wise voting network for 6DoF pose estimation. 2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp 4556–4565. https://doi.org/10.1109/CVPR.2019.00469
44. Poirson P, Ammirato P, Fu C-Y, Liu W, Kosecká J, Berg AC (2016) Fast single shot detection and pose estimation. 2016 Fourth Int. Conf. 3D Vis. 3DV, pp 676–684. https://doi.org/10.1109/3DV.2016.78
45. Rad M, Lepetit V (2017) BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. 2017 IEEE Int. Conf. Comput. Vis. ICCV, pp 3848–3856. https://doi.org/10.1109/ICCV.2017.413
46. Rambach J, Deng C, Pagani A, Stricker D (2018) Learning 6DoF object poses from synthetic single channel images. 2018 IEEE Int. Symp. Mix. Augment. Real. Adjun. ISMAR-Adjun, pp 164–169. https://doi.org/10.1109/ISMAR-Adjunct.2018.00058
47. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, pp 779–788. https://doi.org/10.1109/CVPR.2016.91
48. Sahin C, Garcia-Hernando G, Sock J, Kim T-K (2020) A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators. Image Vis Comput 96:103898. https://doi.org/10.1016/j.imavis.2020.103898
49. Su H, Qi CR, Li Y, Guibas LJ (2015) Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. 2015 IEEE Int. Conf. Comput. Vis. ICCV, pp 2686–2694. https://doi.org/10.1109/ICCV.2015.308
50. Su Y, Rambach J, Pagani A, Stricker D (2021) SynPo-Net—accurate and fast CNN-based 6DoF object pose estimation using synthetic training. Sensors 21:300. https://doi.org/10.3390/s21010300
51. Sundermeyer M, Marton Z-C, Durner M, Triebel R (2020) Augmented autoencoders: implicit 3D orientation learning for 6D object detection. Int J Comput Vis 128:714–729. https://doi.org/10.1007/s11263-019-01243-8
52. Tanzi L, Piazzolla P, Vezzetti E (2020) Intraoperative surgery room management: a deep learning perspective. Int J Med Robot Comput Assist Surg MRCAS 16:1–12. https://doi.org/10.1002/rcs.2136
53. Tanzi L, Vezzetti E, Moreno R, Aprato A, Audisio A, Massè A (2020) Hierarchical fracture classification of proximal femur X-Ray images using a multistage deep learning approach. Eur J Radiol 133:109373. https://doi.org/10.1016/j.ejrad.2020.109373
54. Tanzi L, Piazzolla P, Porpiglia F, Vezzetti E (2021) Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance. Int J Comput Assist Radiol Surg 16:1435–1445. https://doi.org/10.1007/s11548-021-02432-y
55. Tekin B, Sinha SN, Fua P (2018) Real-Time Seamless Single Shot 6D Object Pose Prediction. 2018 IEEECVF Conf. Comput. Vis. Pattern Recognit, pp 292–301. https://doi.org/10.1109/CVPR.2018.00038
56. Tjaden H, Schwanecke U, Schömer E (2017) Real-time monocular pose estimation of D objects using temporally consistent local color histograms. 2017 IEEE Int. Conf. Comput. Vis. ICCV, pp 124–32. https://doi.org/10.1109/ICCV.2017.23
57. Trabelsi A, Chaabane M, Blanchard N, Beveridge R (2021) A pose proposal and refinement network for better 6D object pose estimation. 2021 IEEE Winter Conf. Appl Comput Vis WACV, pp 2381–2390. https://doi.org/10.1109/WACV48630.2021.00243
58. Ulrich M, Wiedemann C, Steger C (2012) Combining scale-space and similarity-based aspect graphs for fast 3D object recognition. IEEE Trans Pattern Anal Mach Intell 34:1902–1914. https://doi.org/10.1109/TPAMI.2011.266
59. Wang Y, Jin S, Ou Y (2019) A multi-task learning convolutional neural network for object pose estimation⋆. 2019 IEEE Int. Conf. Robot. Biomim. ROBIO, pp 284–289. https://doi.org/10.1109/ROBIO49542.2019.8961594
60. Wang G, Manhardt F, Tombari F, Ji X (2021) GDR-Net: geometry-guided direct regression network for monocular 6D object pose estimation. Proc. IEEECVF Conf. Comput. Vis. Pattern Recognit, pp 16611–16621
61. Wu J, Zhou B, Russell R, Kee V, Wagner S, Hebert M et al (2018) Real-time object pose estimation with pose interpreter networks. 2018 IEEERSJ Int. Conf. Intell Robots Syst IROS, pp 6798–6805. https://doi.org/10.1109/IROS.2018.8593662
62. Xiang Y, Schmidt T, Narayanan V, Fox D (2018) PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. Robot. Sci. Syst. XIV, Robotics: Science and Systems Foundation; https://doi.org/10.15607/RSS.2018.XIV.019
63. Xu BChen Z (2018) Multi-level Fusion Based 3D Object Detection from Monocular Images. 2018 IEEECVF Conf. Comput10.1109/CVPR.2018.00249Xu B, Chen Z (2018) Multi-level Fusion Based 3D

Object Detection from Monocular Images. 2018 IEEECVF Conf. Comput. Vis. Pattern Recognit., pp 2345–2353. https://doi.org/10.1109/CVPR.2018.00249
64. Yang Z, Yu X, Yang Y (2021) DSC-PoseNet: learning 6DoF object pose estimation via dual-scale consistency. Proc. IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp 3907–3916
65. You J-K, Hsu C-CJ, Wang W-Y, Huang S-K (2021) Object pose estimation incorporating projection loss and discriminative refinement. IEEE Access 9:18597–18606. https://doi.org/10.1109/ACCESS.2021.3054493
66. Zakharov S, Shugurov I, Ilic S (2019) DPOD: 6D pose object detector and refiner. 2019 IEEECVF Int. Conf. Comput Vis ICCV , pp 1941–1950. https://doi.org/10.1109/ICCV.2019.00203
67. Zhang X, Jiang Z, Zhang H (2019) Real-time 6D pose estimation from a single RGB image. Image Vis Comput 89:1–11. https://doi.org/10.1016/j.imavis.2019.06.013
68. Zhao Z, Peng G, Wang H, Fang H-S, Li C, Lu C (2018) Estimating 6D pose from localizing designated surface keypoints. ArXiv181201387 Cs
69. Zhao W, Zhang S, Guan Z, Luo H, Tang L, Peng J et al (2020) 6D object pose estimation via viewpoint relation reasoning. Neurocomputing 389:9–17. https://doi.org/10.1016/j.neucom.2019.12.108
70. Zhao W, Zhang S, Guan Z, Zhao W, Peng J, Fan J (2020) Learning deep network for detecting 3D object keypoints and 6D Poses. 2020 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pp 14122–14130. https://doi.org/10.1109/CVPR42600.2020.01414
71. Zhu Y, Wan L, Xu W, Wang S, ASPP-DF-PVNet (2021) Atrous spatial pyramid pooling and distance-filtered PVNet for occlusion resistant 6D object pose estimation. Signal Process Image Commun 95:116268. https://doi.org/10.1016/j.image.2021.116268
72. Zou W, Wu D, Tian S, Xiang C, Li X, Zhang L (2021) End-to-End 6DoF pose estimation from monocular RGB images. IEEE Trans Consum Electron 67:87–96. https://doi.org/10.1109/TCE.2021.3057137
73. Zuo G, Zhang C, Liu H, Gong D (2020) Low-quality rendering-driven 6D object pose estimation from single RGB image. Int. Jt. Conf. Neural Netw. IJCNN, 2020, pp 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207286