Review

A Review of Markerless Pose Estimation Methods for Digital Twins

Maximillian K Panoff , Antonio Hendricks , Peter Forcha * , Jung Minchul , Shuo Wang * , Christophe Bobda *

Posted Date: 5 February 2026

doi: 10.20944/preprints202602.0443.v1

Keywords: digital twin; Pose Estimation; computer vision; review

 doi:10.20944/preprints202602.0443.v1

Article

A Review of Markerless Pose Estimation Methods for Digital Twins

Maximillian Panoff, Antonio Hendricks, Peter Forcha *, Minchul Jung, Shuo Wang * and Christophe Bobda *

University of Florida;
* Correspondnece: peter.forcha@ufl.edu (P.F.); shuo.wang@ece.ufl.edu (S.W.); cbobda@ece.ufl.edu (C.B.)

Abstract

As robots and other forms of intelligent automation are improved, solutions to safely and accurately control their operation have also evolved. Many recent solutions to this control problem rely heavily upon Digital Twins (DT), or a digital copy of the environment, which tracks the presence and location of all agents and objects. However, implementing a digital twin is a non-neglible challenge as a high fidelity link between the real and digital worlds must be maintained. To this end, we have compiled a brief background on the problem of 3-Dimensional (3D) Pose Recovery from markerless visual data, listing some of the most recent works in the field and comparing their suitability for DT. In 3D Pose Recovery (3DPR), a subject's position and orientation, or pose, is recovered from visual data in which objects and actors are not physically tagged or identified. This pose can then be used to help identify the location of a target object or actor to track in the digital twin. Additionally, we identify some emerging or remaining challenges in visual 3DPR in the context of DT and highlight a few methods that attempt to address them.

Keywords: digital twin; Pose Estimation; computer vision; review

Note to Practitioners: This document is focused on a comparison between recent solutions for creating digital twins. This technology simplifies and automates the tracking of inventory, parts, and workers throughout a work zone. Digital twins can also be used for more advanced analysis and responsive control of a system. This document focuses on clarifying the strengths and weakness of each approach, in terms of robustness, accuracy, throughput, and complexity of integration.

1. Introduction

The fourth industrial revolution provided significant opportunities for companies to expand workload automation with additional intelligence. Digital Twinning is one of the main methods used to leverage the benefits of these new intelligent systems. In Digital Twinning, a physical work environment is replicated digitally, allowing for additional analysis and more informed decision-making. Many emerging works in task and workload optimization [1–3] rely upon the system having complete knowledge of all the Actors and Objects it contains, known as a *state*. One of the most effective ways to create and maintain this system state is through Digital Twins.

However, creating a Digital Twin can be a significant undertaking for multiple reasons. Firstly, it must be able to operate in real-time, as it provides critical information to the system controller(s), lest it become a bottleneck for overall system throughput. Secondly, Digital Twins must be accurate. If a system controller receives flawed information, it may be less efficient and even cause damage to its components or harm to nearby humans. Finally, a system's state often consists of many different types of components, including both Object and Actor existences and locations. A Digital Twin must be able to capture all relevant information.

 doi:10.20944/preprints202602.0443.v1

This second criteria can be especially challenging, as even with an accurate computer model for a system component, placing that model and maintaining that its position mirrors or 'twins' its real world counterpart requires the position of it's real world counterpart to be known. Pose Estimation includes a family of techniques for estimating the pose of real world objects, making it an essential component of DT. Pose Estimation can make use of many underling sensing technologies, but visual pose estimation, which uses cameras, is an especially active field, due to many recent advances in Deep Learning for Computer Vision. Cameras also provide a non-intrusive way to gather data from environments and, as a result, are one of the most popular and cost-effective sensor types for 3D Pose Recovery (3DPR) available.

3DPR has greatly benefited from the advances made in computer vision, particularly in advances using Deep Learning (DL) and Convolutional Neural Networks (CNNs), especially those for object detection and image segmentation. Works such as YOLO [4], ResNet [5], and the VGG family [6] are highly adaptable to a wide variety of specific tasks in computer vision and allow for basic *Object* (*i.e.* inanimate) recognition and localization while OpenPose [7] and DeepCut [8] can recover *Actor* (*i.e.* animate, self-propelled) poses in 2D. These detections can then be transferred to 3D in several ways, but new and novel techniques like DREAM [9] and PoseCNN [10] are capable of detecting poses in 3D natively.

In this paper, we compile the latest methods of 3DPR, which can be used to create Digital Twins of both Objects and Actors. We additionally introduce several axes for both Object Pose Estimation (OPE) and Actor Pose Estimatio (APE) to describe and compare works within each field. We compare the efficacy of these solutions within our proposed families and, importantly, define and introduce the metrics used for comparison to assist future researchers in better understanding this increasingly important area.

The contributions of this paper are as follows:

- A collection of the latest works in markerless visual 3D Pose Recovery and comparisons between them
- A technological framework to easily describe the capabilities of 3DPR methodologies and their capabilities
- Identification of remaining problems in the field and recent techniques that may assist in addressing them

The rest of the paper is organized as follows: A brief description of some required background knowledge, focusing on performance metrics and relevant datasets, is presented in Section 2. In Section 3 we discuss recent works and propose taxological axes to group them. Following that, in Section 4, we discuss existing challenges and limitations in 3DPR for robotics and recent works focusing on them. Finally, we conclude our paper in Section 5.

2. Background

2.1. Related Works

We begin by introducing a few recent, related survey and identifying the gap that this document seeks to address. A brief summary of this discussion can be found in Table 1. While there are a large number of surveys on many aspects of DT, we have found comparatively few that examine the particular problem of estimating the position of physical objects to ensure that the digital twin is kept up-to-date. This survey is the first to our knowledge to seek to provide an overview of this problem, and a discussion and comparison of techniques to solve it using visual or image based data.

 doi:10.20944/preprints202602.0443.v1

Table 1. A brief comparison of how the contributions of this survey differs from recent related surveys

Survey	Focus and Contribution
Huang *et al.* 2021 [11]	A review of AI enhanced DT rather than how DTs are implemented
Mihai *et al.* 2022 [12]	Holistic coverage of DT, including market impact and computational surfaces, but without discussion on 3DPR.
Silva *et al.* 2022 [13]	Control Techniques using information from DT to improve human safety
Thelen *et al.* 2022 [14]	A comprehensive overview of DT creation, from model creation to state estimation under noisy conditions, but limited discussion of the sensors and sensing techniques used in "Physical-to-virtual (P2V) twinning"
This survey	A dedicated review of visual 3DPR methods to solve the problem of maintaining the physical-virtual link in DT

Pose Estimation (PE) is an established topic with a large body of prior work dedicated to it. Many methods exist to test and quantify success in PE, with many recent approaches building on and expanding those that came before. To begin our analysis of PE for Digital Twins, we must first understand exactly how PE methods are evaluated. To that end, we will introduce current metrics and datasets for PE, especially those relevant to robotics as it is one of the most common use cases for Digital Twinning, and discuss their specifics.

2.2. Metrics for Pose Estimation in Robotics

We will begin by examining the metrics commonly used in pose estimation for actors. As mentioned in Section 1, we propose breaking down the problem of PE into two subcategories: Actor Pose Estimation (APE) and Object Pose Estimation (OPE). Of course, many potential metrics exist for Actor Pose Estimation, such as Percentage of Detected Joints (PDJ) [15], Percentage Correct Keypoints (PCK) [16], Percentage of Correct Parts (PCP) [17], and others, but these do not apply to OPE. We will begin by covering the metrics that hold for both OPE (*i.e.*, for objects) and APE (*i.e.*, for actors). A comparison of all metrics can be found in Table 2.

Table 2. A comparison between different 3D pose estimation metrics used in robotic applications. A negative performance gradient means lower values indicate increased performance, and positive the inverse.

Metric Name	**Performance Gradient**	**Best For**	**Notes**
ADD(-S)	Negative	Objects (PEG)	Compares the average distance between each point in the predicted location of a 3D model and the model's true position.
VSD	Negative	Objects (PEG)	Compares the distance between the closest points in 3D surfaces. If the distance is below some threshold, it is ignored, otherwise it adds a flat value irrespective of actual distance.
PDJ	Positive	Actors (RPE)	If the distance from the predicted point to the true is within a percentage of the torso record 1, otherwise 0. Average across all joints.
PCP	Positive	Actors (RPE)	If the distance from the predicted point to the true is within a percentage of the limb length, record 1, otherwise 0. Replaced by PDJ.
PCK	Positive	Actors (RPE)	If the distance from the predicted point to the true is within a percentage of the total area of the subject record 1, otherwise 0.
MPJPE	Negative	Actors (RPE)	Sum all the distances between predicted and true joints for each actor in the scene. It can be scaled by actor size in multiple ways.
MPJAE	Negative	Actors (RPE)	Sum all the distances between predicted and true joints for each actor in the scene divided by the true length of the corresponding limb.

 doi:10.20944/preprints202602.0443.v1

2.2.1. Object Pose Estimation Metrics

These metrics are more commonly encountered in works focusing on Object pose recovery, as individual pose accuracies have more intuitive meaning when dealing with rigid subjects rather than a collection of connected joints that actors have. *I.e.* A single recovered pose does not provide sufficient information for a system to describe a deformable or changeable Actor.

Accuracy of the Average Distance: Accuracy of the Average Distance (ADD) [18] and Accuracy of the Average Distance for Symmetric Objects (ADD-S) [10] are often deployed in conjunction, denoted ADD(-S). Both attempt to quantify the distance of a predicted 3D Object to a known ground truth, with the addition of ADD-S being redefined to accommodate symmetrical Objects better. Symmetrical Objects have long been an issue in OPE, as recovering their rotation from the axis they are symmetric around is often impossible. ADD is described in Equation 1 where m is the number of points in 3D model P, R and T denote the ground truth rotation and translation of that model while $\hat{R}$ and $\hat{T}$ denote the estimated rotation and translation of that model.

$$\text{ADD} = \frac{1}{m} \sum_{x \in P} || \ (Rx + T) - (\hat{R}x + \hat{T}) \ || \tag{1}$$

$$\text{ADD-S} = \frac{1}{m} \sum_{x_1 \in P} \min_{x_2 \in P} || \ (Rx_1 + T) - (\hat{R}x_2 + \hat{T}) \ || \tag{2}$$

Visible Surface Discrepancy: Visible Surface Discrepancy (VSD) is a second approach to deal with the problem of symmetric Objects [19]. In VSD, masks are generated to cover the visual portions of the ground truth Object and the predicted Object. 3D Surfaces are then generated using these masks, and the distance between them is found. This distance is then averaged to create the final VSD metric. Please see Equation 3 for more detail. Note that p is a single pixel in the union of the two masks. S_p denotes the point on the 3D surface of the masked ground truth image corresponding to pixel P and $\hat{S}$ the same for the predicted object. τ is a tolerance parameter. As an error metric, a lower VSD is better.

$$VSD = \frac{1}{P} \sum_{p=0}^{P} \left\{ \begin{array}{ll} \sum_{p=0}^{P} \frac{|\hat{S}_p - S_p|}{\tau}, & \text{if } | \ \hat{S}_p - S_p \ | < \tau \\ 1, & \text{if } | \ \hat{S}_p - S_p \ | >= \tau \end{array} \right\} \tag{3}$$

2.2.2. Actor Pose Estimation Metrics

Unlike Objects, Actors are commonly defined as having several joints that must evaluated concurrently. Additionally, Actors, especially human Actors, are prone to deformation, which complicates evaluations based on 3D models.

Mean Per Joint Position Error: The above metrics all hold for identifying the similarity or distance for individual Objects, but when working with Actors that have many constituent joints, Mean Per Joint Position Error (MPJPE) may be more useful. It is important to note that as MPJPE is an error metric, lower scores are better. Introduced along with the Human3.6M Dataset [20] (Section 2.3.7), MPJPE is a family of metrics that share a similar underlying goal. MPJPE and the similar Mean Per Joint Angle Error (MPJAE) seek to convey how closely all the predicted point locations align with the true joint locations for a single Actor as a single number. Simply put, the 3D error for all joints belonging to an Actor, either positional such as in ADD(-S) (Section 2.2.1) or angular, are summed and then divided by the number of joints that the Actor has, as shown in Equation 4, where N is the number of joints for the Actor, and J_i denotes the true location in $\mathcal{R}^3$ of the Actor's ith joint and $\hat{J}_i$ the same for the estimated joint. Extensions of this normalize error by limb length as well [20]. MPJAE is similarly defined but for joint angles instead.

$$MPJPE = \frac{1}{N} \sum_{i=0}^{N} || \ \hat{J}_i - J_i \ ||^2 \tag{4}$$

 doi:10.20944/preprints202602.0443.v1

There are two obvious issues with this metric. Firstly, as noted by the authors, this metric does not distinguish between a single joint that is very poorly estimated and many joints with a medium amount of error, as the total error will be similar in both cases. Secondly, and of particular concern for robotics, default MPJPE does not control for limb length, which can vary greatly in different robot models. Angular offsets can result in different amounts of positional error, depending on limb length, which is why MPJAE is also an important metric. This can be resolved by dividing the error by the length of the limb and standardizing it. Secondly, errors in offset from each prior limb may not be accounted for. This is to say, if the root of the Actor's skeleton is misaligned, all predicted joints will have some amount of error, even if their position relative to the root matches that in ground truth. This can be resolved by aligning the root of each skeleton or one of any connecting joints when evaluating the metric.

2.3. Datasets and Benchmarks

In this section, we introduce and compare a few datasets commonly used to evaluate the performance of PE models. We pay particular attention to the intent behind choosing to evaluate on these datasets and the data they contain. Choosing a dataset that closely reflects the intended system is critical and highly specific to the system, but as a general rule of thumb, LineMOD/YCB are good choices for system using either household or highly dissimilar (especially color-wise) objects, LineMOD-OCCLUDED if the effect of occlusion needs to be specifically measured, and T-LESS for manufacturing situations. If they system includes transparent objects that need to be twinned, then PhoCal should be used. For human actors Human3.6m is generally preferable to SURREAL, although the latter has the benefit of being synthetic and modifiable for new types of Actors (Non-Human).

2.3.1. YCB-Video (Objects)

Introduced by PoseCNN [10], this real-world dataset consists of Objects placed directly on different surfaces. Includes RGB-D images, segmentation ground truth data, and poses relative to the camera of 21 Objects from the YCB model set [21] across 92 videos with 133,827 frames. YCB-Video also includes camera intrinsics but does not have camera extrinsics. YCB is often used to determine the effectiveness of single-view systems, as it does not support cross-view comparison due to the lack of extrinsics.

2.3.2. YCB-M (Objects)

YCB-M is a multi-camera dataset with several YCB-style Object setups captured from many viewpoints by multiple RGB-D cameras. Unlike YCB-Video, it includes camera extrinsics. It also allows techniques to evaluate how different camera types affect performance.

2.3.3. LineMOD (Objects)

Introduced by [22], this real-world dataset consists of Objects placed onto a ChArUco Board with Object pose given relative to the camera. Ground truth RGB-D and segmentation images and camera intrinsic are available, but LineMOD does not contain camera extrinsics directly. Items are placed onto an ArUco [23] board, which can be used to calculate camera extrinsics. LineMOD is currently the most common choice for OPE evaluation.

2.3.4. LineMOD-OCCLUDED (Objects)

A specific subset of LineMOD featuring heavy occlusion by a large table clamp. It is often used as a standalone dataset to evaluate model performance in occlusion-heavy conditions.

2.3.5. PhoCal (Objects)

PhoCal [24] provides a comprehensive benchmark for OPE, with challenges and datatypes not included in any other dataset to date. Specifically, it includes transparent and reflective Objects, which

 doi:10.20944/preprints202602.0443.v1

challenge modern OPE solutions. Additionally, its test set has the same categories or types of items but not the same instances, thus providing a better test of model generalization than prior Object datasets.

2.3.6. SURREAL (Actors)

SURREAL [25] consists of over 6 million frames of simulated humans performing different actions under numerous lighting conditions and backgrounds. The publicly available dataset includes camera intrinsics and extrinsics, ground truth 2D and 3D poses, and pixel-level limb segmentation for many Actor Classes.

2.3.7. Human3.6M (Actors)

The Human3.6M dataset [20] consists of 3.6 million 3D human pose and RGB image pairs taken from 4 views. Each RGB image also has semantic segmentation masks for 24 body parts, background removal, and precomputed bounding boxes. Eleven human Actors had complete 3D body scans taken and participated in seventeen activities/scenarios. Human3.6M is often used to determine the effectiveness of APE.

2.3.8. T-LESS (Objects)

T-LESS provides RGB-D captures in a setting similar to LineMOD (*i.e.* multiple objects on an ArUco board) but instead uses 30 industry-relevant objects with low textures and similar shapes for enhanced challenge [26]. 3 types of sensors (structured light, time of flight, and RGB) were used to collect 38k training and 10k test images.

2.3.9. CAMERA (Objects)

Context-Aware MixEd ReAlity (CAMERA) refers to the synthetic portion of the Normalized Object Coordinate Space (NOCS) dataset which combines 31 (mostly tabletop) real RGB-D scenes with computer models of 1000+ objects belonging to 6 object categories (bottle, bowl, camera, can, laptop, and mug) synthetically inserted to create 300k images. [27]. It has a designated validation set of 25k images, leading the dataset to be referred to as CAMERA25, but it is important to note that CAMERA25 refers to the complete 300k image dataset, not only the 25k validation set.

2.3.10. REAL275 (Objects)

REAL275 is the real-world complement to the CAMERA dataset, so named for the 275 test images. Unlike CAMERA, this dataset comprises entirely of real world images captured in 8K RGB-D with the same 18 scenes used in CAMERA but with 42 real objects [27].

3. Deep Learning for 3DPR

In Section 2, we introduced metrics and datasets used for evaluation, providing the necessary foundation for a discussion on current works on PE. In this Section, we discuss the state-of-the-art in PE, group approaches into families based on the details of their approach, and set the stage for our identification of challenges in Section 4.

3.1. Object Pose Estimation

We propose Object Pose Estimation (OPE) as a family of solutions to acquire Object poses for Digital Twins from visual data. We identify recent works in this field and compare them using standard and novel calculated metrics (Table 3).

 doi:10.20944/preprints202602.0443.v1

Table 3. A comparison between Object based 3DPR methods. Mean Error Area Under Curve (AUC) Accuracy of the Average Distance for the closest object (ADD-S) or as reported by the work.Throughput is noted in Frames Per Second (FPS). †Time to complete inference includes robot arm movement. The best value in each column is bolded.

Technique	RGB(D)	YCB [21]	LineMOD [22]	LM-Occluded [28]	MOR	TI	IC	FPS	GPU
PoseCNN 3.1.2.1	RGB	75.9%	–	24.9%	2	4	8	–	–
PoseCNN 3.1.2.1	RGB-D	93.0%	–	78.0%	2	4	8	–	–
DOPE 3.1.2.2	RGB	80.5%	–	–	**3**	3	3	4.31	NVIDIA Titan X
Pix2Pose 3.1.2.3	RGB	–	72.4%	32.0%	1	3	4	6.5	NVIDIA 1080
Efficient Pose 3.1.2.4	RGB	–	97.25%	**83.89%**	**3**	4	**2**	27.45	NVIDIA 2080Ti
SS6D 3.1.2.5	RGB-D	60.8%	–	–	2	2	Iter	0.1†	2×NVIDIA Titan X
Self6D++3.1.2.6	RGB	90.5%	85.6%	59.8%	1	2	Iter	100	NVIDIA Titan X
Self6D++3.1.2.6	RGB-D	91.1%	88.5%	64.7%	1	2	Iter	25	NVIDIA Titan X
RePose 3.1.2.7	RGB	70.5%	96.1%	51.6%	1	2	Iter	91	NVIDIA 2080 Super
SO-Pose 3.1.2.8	RGB	90.9%	96%	62.3%	1	4	7	20	NVIDIA Titan X
FFB6D 3.1.2.9	RGB-D	92.7%	**99.7%**	66.2%	2	4	4	13.3	–
ROPE 3.1.2.10	RGB	79.88%	95.61%	45.95%	1	4	6	–	–
Template Pose 3.1.2.11	RGB	–	99.1%	79.4%	2	3	3	125	NVIDIA V100
OVE6D 3.1.2.12	RGB-D	–	92.4%	72.8%	1	3	6	20	NVIDIA RTX3090
OSSID 3.1.2.13	RGB-D	63.2%	–	64.0%	1	2	Iter	4.34	–
OSOP 3.1.2.14	RGB	80%	86%	61%	1	3	4	–	–
Gen6D 3.1.2.15	RGB	–	20.74%	–	2	4	Iter	128	NVIDIA 2080Ti
OnePose 3.1.2.16	RGB-D	–	76.9%	–	1	**1**	3	12.5	NVIDIA V100
TexPose 3.1.2.17	RGB	–	91.7%	66.7%	1	3	4	–	–
ICG+ 3.1.2.18	RGB	**94%**	97.7%	–	1	**1**	3	**312.4**	NVIDIA RTX A5000
CASAPose 3.1.2.19	RGB	–	68.1%	35.9%	**3**	3	4	27	NVIDIA RTX A100

 doi:10.20944/preprints202602.0443.v1

3.1.1. Axes

We propose several independent axes, namely 'Multi-Object Robustness,' 'Training Intensity,' and 'Inference Complexity,' to group recent works and differentiate between them. This provides a more intutive summary of technique characteristics compared to traditional taxological trees. Table 4 briefly overviews the meaning of these axes.

Table 4. The axes of Object Pose Estimation values. *Solutions that iterate are marked by 'Iter'; otherwise, the IC score is the same as the number of logical steps or stages.

Axis	Level 1	Level 2	Level 3	Level 4
Multi-Object Robustness (MOR)	1 Object, 1 Class	1 Object, 2+ Classes	2+ Object, Unique Classes	2+ Object, 2+ Classes
Inference Complexity (IC)	1 Stage	2 Stages	3 Stages	4 Stages*
Training Intensity (TI)	Unsupervised	Semi-Supervised	Supervised Transferable / Synthetic	Supervised
Multi-Agent Robustness (MAR)	1 Actor, 1 Class	1 Actor, 2+ Classes	2+ Actors, Unique Classes	2+ Actors, 2+ Classes

Multi-Object Robustness

Multi-Object Robustness (MOR) axis captures the proposed system's ability to handle concurrent Objects in the scene. As many components must be detected and registered simultaneously in Digital Twinning, the ability to handle multiple Objects and Classes of Objects is paramount. Level 1 systems can only detect a single instance of an Object belonging to a single Class. Level 2 systems can find single instances of multiple Objects concurrently. Levels 1 and two may be able to handle multiple instances through an early Region of Interest (RoI) selector, which allows the actual processing to work on a small area of data predetermined to contain a single instance. Level 3 systems can natively handle multiple instances of the same Object Class, and level 4 systems can natively handle multiple instances of multiple Object Classes concurrently.

Inference Complexity

Inference Complexity (IC) represents how many processing stages and overhead a technique has before outputting final results. For example, ROPE [29] is 'Indirect' as it identifies Regions of Interest (RoIs), extracts 2D features from those RoIs, and aligns point clouds with those features before finally extracting pose information. Conversely, EfficientPose [30], which extracts rotational and translational information in a single stage, is 'Direct.' Note that we define stages differently than many prior works in this area. Specifically, each point in a pipeline at which the result can be interpreted to have an explicit meaning different from the prior stages is identified as a stage, *i.e.* each logical step in a process is a stage. IC is the total number of such stages or 'Iter' if the process includes any arbitrary number of iterations. It is important to note that point cloud registration (*i.e.* aligning one point cloud onto another) is almost exclusively iterative, and so any technique using such a process will be iterative as well. Most traditional solutions to 3DPR involve registration, and one of the main advantages of visual Deep Learning based 3DPR is the ability to skip or mitigate this process.

Training Intensity

Finally, Training Intensity (TI) concerns the level of supervision and data cost needed to train a DL model. These qualities are critical to determining the difficulty of actual implementation. Self6D++ [31], which is trained through self-supervision, would be more 'Relaxed' than PoseCNN [10], which follows a traditional supervised learning structure on a large dataset, would be 'Intense.' The higher the TI, the more work it would take to train a model. Level 1 is an unsupervised or zero-shot approach, 2 includes semi-supervised or few-shot methods, 3 has solutions that include synthetic or highly transferable datasets, and 4 contains fully supervised pipelines.

3.1.2. Works

We compile and compare several works on OPE which are shown in Table 3 and Figure 1. More detailed descriptions for each work are available below.

 doi:10.20944/preprints202602.0443.v1

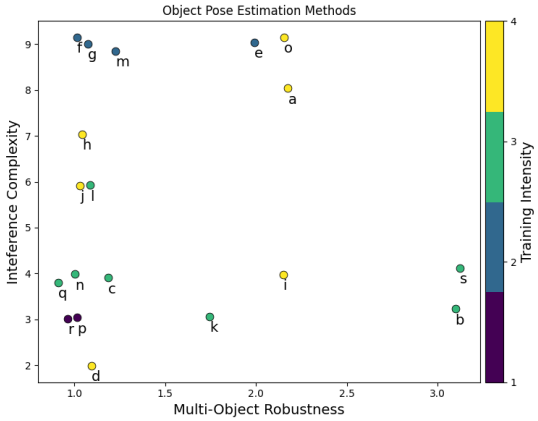


Figure 1. A plot of recent OPE methods along the three axis specified, color being Training Intensity. Any given method denoted by 'X' can be found in Section 3.1.2X.

PoseCNN

One of the first OPE solutions, PoseCNN [10]is a methodology that uses multiple stages to place 3D points in a scene virtually. An Object's pose is then recovered through this virtual point cloud. PoseCNN supports multiple concurrent objects with an (RoI) pre-selection stage.

DOPE

Deep Object Pose Estimation (DOPE) [32] uses virtual data in a supervised fashion when training to mitigate the difficulty of acquiring data. It converts recovered 2D landmarks to 3D centroid points through the Perspective and Point (PnP) algorithm [33].

Pix2Pose

[34] is a technique that uses multiple stages to predict the pose of a 3D point cloud or model. It uses an Autoencoder-based General Adversarial Network (GAN) to predict 3D clouds from input images, with the discriminator attempting to differentiate ground truth textureless renderings and training data.

EfficentPose

An 6D pose estimation approach focused on efficiency, EfficentPose [30] utilizes the EfficientNet model as the backbone for multiple object detection. The framework introduces rotational and translation prediction subnets following a backbone for 6D object pose estimation. EfficentPose uses a novel 6D augmentation method to ensure generalization.

SS6D

Deng *et al.* present a novel approach in robot manipulation, employing a self-supervised learning framework for 6D object pose estimation [35]. This method's self-supervision is a significant feature, using a robot arm to help with the estimation processes, thereby reducing the method's dependence on large-scale, labeled data.

Self6D++

This monocular self-supervised 6D pose estimation approach uses both synthetic data and unannotated real images; Self6D++ [31] leverages neural rendering to ensure compliance of objects with geometrical constraints. Self6D++ also uses self-supervision to increase flexibility.

RePOSE

This method involves mapping 3D model textures to a 2D projection and then using Levenberg-Marquardt optimization to minimize the distance between the input and rendered images [36].

 doi:10.20944/preprints202602.0443.v1

SO-POSE

This architecture directly regresses the 6D pose from each object's two-layer representation (2D-3D point matching, self-occlusion information) [37]. This two-layer representation, which combines 2D-3D correspondences with self-occlusion information, is used to enforce cross-layer consistencies, producing an improved accuracy for OPE.

FFB6D

A novel bidirectional fusion network designed for 6D pose estimation from a single RGBD image [38]. It proposes novel bidirectional fusion modules that bridge CNN and Point Cloud networks. This allows the networks to utilize local and global information from each other for better appearance and geometry representation learning.

ROPE

ROPE [29] focuses on the connections and coherence between 2D landmarks through Multi-Precision Supervision (MPS). This MPS is then enhanced through a novel Occlude-and-blackout Batch Augmentation (OBA) training approach to promote robust prediction under occlusion.

Template-Pose

Template-Pose [39] is a method using Contrastive Learning to create an embedded space in which different classes of objects are maximally distant from each other. This embedded space is also used to determine approximate occupancy in 3D space, allowing Template-Pose to recover 3D models of objects excluded from training and partially occluded simultaneously.

OVE6D

OVE6D [40] uses depth images and RGB data to calculate object pose. Specifically, many viewpoints are rendered during training to generate synthetic data, then used to create a 'reference codebook.' During inference, this codebook provides starting points for a regression solver to identify poses with similar content to the input data.

OSSID

OSSID aims to enhance object pose estimation through a zero-shot pose estimator to self-supervise the training of a fast detection algorithm [41]. This moderately intense method significantly improves the inference speed of pose estimation from live video-feed systems by using the fast, continually recursive detector to filter input for the pose estimator, enabling efficient and accurate pose estimation without manual annotations.

OSOP

[42] utilizes a four-step pipeline: identification of an object using a single image based on a pre-existing 3D model, initial viewpoint estimation by template matching, dense 2D-2D matching between the selected image segment and the corresponding template, and finally, 6 DoF pose estimation using PnP+RANSAC or Kabsch+RANSAC. This approach generalizes to new objects without prior training.

Gen6D

A framework for 6D pose estimation of novel objects using only RGB images, Gen6D incorporates an image-matching procedure for step-by-step pose estimation in a coarse-to-fine manner via an object detection module, a mechanism for selecting viewpoints and a pose refining stage [43]. It shows comparable performance to specialized pose estimation models on benchmarks such as LINEMOD.

OnePose

Sun *et al.* propose using a two-phase approach: an offline mapping phase to build a sparse Structure from Motion (SfM) model via a full video scan of the object, and an online localization phase

 doi:10.20944/preprints202602.0443.v1

employing a graph attention network (GAT) for direct 2D-3D feature matching along with a PnP solver to maintain object pose localization accuracy [44]. This technique allows robust pose estimation adaptable to significant viewpoint, lighting, and scale variations.

TexPose

[45] is a self-supervised learning OPE framework for 6-DoF Object Pose Estimation centered on learning process decomposition into texture and pose components. The method leverages texture learning from real image collections and pose estimation from synthetically generated, geometrically accurate data. This dual approach allows for creating photorealistic training views and facilitates pose estimator supervision with precise geometry and appearance.

ICG+

Iterative Corresponding Geometry expanded [46] is a multi-modality tracker that fuses information from visual appearance and geometry to estimate object poses. The algorithm extends a previous method, Iterative Corresponding Geometry (ICG) [47]. It combines local characteristics that describe distances between landmarks in keyframes and in the current image and global differences along the object's surface.

CASAPose

CASAPose uses 2D segmentation to predict vector fields, which are then aligned with 3D models through PnP. Importantly, it is one of the few methods to support concurrent instances of multiple targets [48].

3.2. Actor Pose Estimation

3.2.1. Axes

Actor Pose Estimation (APE) shares two axes, Training Intensity and Inference Complexity, with OPE, defined in Sections 3.1.1.3 and 3.1.1.2, respectively. However, instead of Output Density, APE's third axis is Multi-Agent Robustness (MAR), described below.

Multi-Agent Robustness

Multi-Agent Robustness refers to the method's ability to function when more than one actor is in the scene. It has four distinct levels. At the most basic, an APE method only supports one active actor and one actor class. Level 2 expands this to include multiple classes of actors, but still only one at a time. Levels 3 and 4 add support for multiple simultaneous actors, with 3 being limited to a single class of actors and 4 supporting multiple types.

3.2.2. Works

We compile and compare several works on APE which are shown in Table 5 and Figure 2. More detailed descriptions for each work are available below.

 doi:10.20944/preprints202602.0443.v1

Table 5. A comparison between Robot Actor-based 3DPR methods. Mean error in cm, except for CASA, which is only recorded as mean Chamfer Distance.

Technique	Mean Error (cm)	MAR	TI	IC
CRAVES 3.2.2.1	2.67	1	3	3
DREAM 3.2.2.2	2.74	1	3	2
Anipose (Human, 90^{th}percentile) 3.2.2.3	8.6	4	4	4
RoboPose 3.2.2.4	1.34	1	3	Iter
Watch-it-move 3.2.2.5	7.59	1	2	6
SPDH 3.2.2.6	4.41	1	3	3
SGTAPose 3.2.2.7	1.812	1	3	3
CASA 3.2.2.8	0.053 (mCham)	1	1	Iter

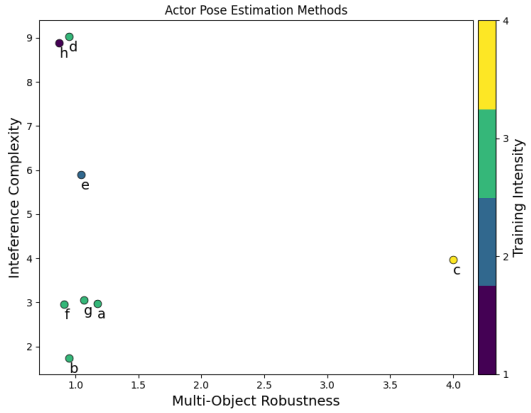


Figure 2. A plot of recent APE methods along the three axis specified, color being Training Intensity. Any given method denoted by 'X' can be found in Section 3.2.2.

CRAVES

The "CRAVES" paper by Y. Zuo *et al.* [49] introduces a semi-supervised APE method using synthetic data from a 3D model and domain adaptation. In terms of our proposed taxonomy, it is 'Sparse' in Density as it focuses on pose estimation without full scene reconstruction, 'Indirect' in Directness due to its multi-stage process, including synthetic data training and real-world adaptation, and 'Moderate' in Intensity because of its iterative optimization algorithm and geometric constraints among keypoints.

DREAM

In a pursuit akin to making a robotics dream a reality, T. E. Lee *et al.* introduce a deep neural network method to detect 2D projections of robot keypoints from a single RGB image, employing Perspective-n-point (PnP) for camera pose estimation [9].

Anipose

Anipose [50] is an open-source Python toolkit for 3D animal pose estimation built on the 2D tracking method DeepLabCut and includes four main components: a 3D calibration module, filters for 2D tracking errors, a triangulation module integrating temporal and spatial regularization, and a structured processing pipeline for video data.

 doi:10.20944/preprints202602.0443.v1

RoboPose

Labbé *et al.* introduce RoboPose [51], a method for estimating the joint angles and the 6D camera-to-robot pose from a single RGB image of a known articulated robot. RoboPose uses a render & compare approach, where it renders a robot image from an estimated state and iteratively refines this state to match the observed robot in the input image. The method can be trained on synthetic data and generalized to unseen robot configurations

Watch It Move

Watch it Move [52] learns the appearance and the structure of articulated objects by observing their movement from multiple views without requiring any annotations or prior knowledge about the structure. Object parts are modeled as ellipsoids before combining their explicit and implicit representations. This method efficiently identifies joints, enabling the re-posing of objects in unseen configurations by rotating and translating each part around its joint.

SPDH

Simoni *et al.* introduce an APE method called Semi-Perspective Decoupled Heatmaps (SPDH) [53], leveraging depth data to predict bi-dimensional heatmaps, which are then converted into 3D joint locations. This method does not require direct access to the robot's internal state information and is trained on synthetic data for seamless real-world deployment. This method balances the characteristics of the heatmap and depth map approaches.

SGTAPose

SGTAPose [54] leverages a temporal cross-attention strategy for efficient feature fusion of successive frames and employs a Perspective-n-point (PnP) solver for initial pose estimation, which is further refined using robot structure priors. The main novelty of SGTAPose is its use of consecutive frames from a single viewpoint. Given its complex use of temporal information and structure priors for enhanced pose estimation, this method is less direct than DREAM.

CASA

[55] Not to be confused with CASAPose, Wu *et al.* employ a two-step process for reconstructing the skeletal shapes of animals from monocular videos called CASA. It first retrieves a shape from a 3D character asset bank using a language-vision model, then uses an inverse graphics framework to infer deformation, skeleton structure, and skin.

3.3. Selection Criteria

Choosing a 3DPR solution is a complex and intensive process, with the ideal solution depending on many aspects of the system such as: Data Availability, Timing Requirements, and Subject Diversity. These constraints, especially data availability and subject complexity are well captured by the calculated metrics used in this paper: MOR/MAR, TI, and IC.

3.3.1. Data Availability

Having access to or the ability to generate large amounts of labeled data greatly expands the range of potential solutions, as well as solution performance. This level of data quality is needed by solutions with a TI of 4. However sometimes, time is abundant and 3D models of the subjects are available but actual data is not, which is often when synthetic data (TI of 3) is used. Other times further data cannot be collected (perhaps due to a lack of access to models or environment), and labeling the data is expensive, which is an ideal scenario for solutions with a TI of 2. Finally, perhaps one does not know all the objects that need to be included in the system before hand, and thus *a priori* data cannot exist, necessitating unsupervised (TI of 1) models.

 doi:10.20944/preprints202602.0443.v1

3.3.2. Subject Diversity

Subject complexity refers to the range and number of subjects that a DT system is attempting to maintain twins of. The ability of each solution along these lines is well captured by the MOR and MAR metrics for Object and Actor subjects, respectively.

3.3.3. Timing Requirements

Unlike Data Availability and Subject Diversity, timing requirements are not wholly captured by our calculated metric, IC. This is due to processing time for 3DPR solutions being heavily dependant on the hardware that is doing the processing. IC provides a 'ballpark' metric on how long solutions may take relative to one another, but not if a particular solution will meet requirements for a particular system. To determine this, a solution should be implemented on the final hardware and the processing time recorded. At the same time, the amount of 'buffer' time before an error or missed update of a twinned component resulting in a hazard should be calculated. If the 3DPR processing time (and the time of any additional required actions) is less than the buffer time before system failure, the 3DPR solution may be satisfactory for the system [56][1].

4. Future Work

While a large accomplishments have been made in both DT and computer vision solutions for 3DPR, we have found surprising little work done on novel visual 3DPR solutions explicitly designed for a generalized DT task. As mentioned in Section 1, DT has several unique challenges and constraints for 3DPR that expand upon the standard problem and require novel solutions. These particular areas of improvement are discussed in this section, along with brief introductions of techniques from 3DPR that at least partially address them, although without particular consideration to the DT space.

4.1. Multi-Actor, Multi-Object Scenes

Even the most recent works in PE typically focus on a single instance [52–55]. While this is an important step in digital twinning, it may not be easily expanded to an environment where multiple actors and are present in a scene, which is likely more common than a single instance in practice.

Most recently, this problem has been taken up by the *Anipose* [50] framework for APE or CAS-APose, DOPE, and EffecientPose [30,32,48] frameworks for OPE. However, most of these methods operate at with MOR/MAR capabilities we define as 'Level 3,' which means that they may not be able to address multiple instances of the same class simultaneously. Of those that do, the solutions tend unsuited for real-time use.

4.2. Occlusion

Occlusion refers to a scenario in which the camera does not have a clear view of the scene or subject due to some obstruction between the camera and the subject. Occlusion is a challenge particularly relevant to monocular (single-view) 3D PE but not exclusive to it. Indirect methods that extract 2D features have native mitigation against this challenge as such solutions only need to recover enough features to align a 3D model but often have large overheads.

A subset of Occlusion that has received minimal evaluation is what we propose calling Same-on-Same occlusion, where two Objects or Actors of the same class occlude each other. This scenario is challenging even for multi-view solutions, which typically resist other forms of occlusion due to the benefit of multiple perspectives.

4.3. Comprehensive Multi-Robots as Actors Datasets

To our knowledge, no publicly available PE datasets use multiple robot Actors. Some datasets include human poses with robots visible in the scene, such as InHARD [57], but only include ground truth pose data for the human. Others include single robots. Not only does this lack inhibit training

[1] This is intended as a general guideline and not specific advice guaranteeing stability in any given system

 doi:10.20944/preprints202602.0443.v1

APE solutions that address Multi-Actor Robustness (MAR, Section 3.2.1.1) and Same-on-Same occlusion described above, it also complicates the fair evaluation of proposed methodologies due to the lack of a standard benchmark. A benchmark or benchmarks would be composed similarly to Human3.6M, with multiple cameras, RBG and RGBD images, activity descriptions, pixel-level segmentation masks, 3D point clouds of the robots, and 2D and 3D poses. Ideally, it would also include multiple types of robot actors (e.g., humanoid, arm, etc.) in addition to multiple form factors (*i.e.*, Kinova Gen3, Franka Panda, etc.) and multiple instances of each factor and include Actor-on-Actor occlusion.

4.4. Pose Estimation Without 3D Object Models

Currently, *a priori* acquisition of 3D computer models of detected objects is required by most solutions in this space. While not overly difficult to acquire, this limitation impacts the flexibility of these solutions. Gen6D attempts to solve this by providing a method for users to essentially create a 3D by taking a short video of the object, but it still fundamentally requires a 3D model for function [43]. LatentFusion instead attempts to create a latent 3D understanding of an object rather than a direct model [58], but it has several limitations itself, especially in inference time and scene complexity. Nevertheless, the approach taken by LatentFusion presents an interesting direction towards PE for digital twinning without explicit 3D models. PIZZA is an alternative to this direction, which explicitly supports unseen objects by estimating 3D changes in position from 2D motion over time.

5. Conclusion

In this paper we provide a background into 3D Pose Estimation and collate a few of the most recent works in the field that can be applied to create Digital Twins. We examine how works evaluated on three recent benchmarks: LineMOD, LineMOD-Occluded, and YCB, compare, in terms of accuracy, throughput, and three novel axes. We note several current remaining questions in the field, such as multi-actor robot pose estimation, and present solutions to those problems based on work in other fields of computer vision.

Funding: The authors of this paper are from the University of Florida's Department of Electrical and Computer Engineering. This material is based upon work supported by the National Science Foundation under Grant Numbers: 2007210 and 2106610.

Acknowledgments: This material is based upon work supported by the National Science Foundation under Grant Numbers: 2007210 and 2106610.

References

1. Alirezazadeh, S.; Alexandre, L.A. Dynamic Task Scheduling for Human-Robot Collaboration. *IEEE Robotics and Automation Letters* **2022**, *7*, 8699–8704. https://doi.org/10.1109/LRA.2022.3188906.
2. Fusaro, F.; Lamon, E.; Momi, E.D.; Ajoudani, A. An Integrated Dynamic Method for Allocating Roles and Planning Tasks for Mixed Human-Robot Teams. In Proceedings of the 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), 2021, pp. 534–539. https://doi.org/10.1109/RO-MAN50785.2021.9515500.
3. Pupa, A.; Van Dijk, W.; Secchi, C. A Human-Centered Dynamic Scheduling Architecture for Collaborative Application. *IEEE Robotics and Automation Letters* **2021**, *6*, 4736–4743. https://doi.org/10.1109/LRA.2021.3068888.
4. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
6. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
7. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7291–7299.

 doi:10.20944/preprints202602.0443.v1

8. Pishchulin, L.; Insafutdinov, E.; Tang, S.; Andres, B.; Andriluka, M.; Gehler, P.V.; Schiele, B. Deepcut: Joint subset partition and labeling for multi person pose estimation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4929–4937.
9. Lee, T.E.; Tremblay, J.; To, T.; Cheng, J.; Mosier, T.; Kroemer, O.; Fox, D.; Birchfield, S. Camera-to-Robot Pose Estimation from a Single Image. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9426–9432. https://doi.org/10.1109/ICRA40945.2020.9196596.
10. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *Robotics: Science and Systems (RSS)* **2018**.
11. Huang, Z.; Shen, Y.; Li, J.; Fey, M.; Brecher, C. A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics. *Sensors* **2021**, *21*. https://doi.org/10.3390/s21196340.
12. Mihai, S.; et al.. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 2255–2291. https://doi.org/10.1109/COMST.2022.3208773.
13. Proia, S.; Carli, R.; Cavone, G.; Dotoli, M. Control Techniques for Safe, Ergonomic, and Efficient Human-Robot Collaboration in the Digital Industry: A Survey. *IEEE Transactions on Automation Science and Engineering* **2022**, *19*, 1798–1819. https://doi.org/10.1109/TASE.2021.3131011.
14. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization* **2022**, *65*, 354.
15. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660. https://doi.org/10.1109/CVPR.2014.214.
16. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686–3693. https://doi.org/10.1109/CVPR.2014.471.
17. Eichner, M.; Marin-Jimenez, M.; Zisserman, A.; Ferrari, V. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International journal of computer vision* **2012**, *99*, 190–214.
18. Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; Navab, N. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In Proceedings of the Asian conference on computer vision. Springer, 2012, pp. 548–562.
19. Hodaň, T.; Matas, J.; Obdržálek, Š. On evaluation of 6D object pose estimation. In Proceedings of the European Conference on Computer Vision. Springer, 2016, pp. 606–619.
20. Ionescu, C.; Papava, D.; Olaru, V.; Sminchisescu, C. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *36*, 1325–1339.
21. Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; Dollar, A.M. Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set. *IEEE Robotics Automation Magazine* **2015**, *22*, 36–52. https://doi.org/10.1109/MRA.2015.2448951.
22. Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; Navab, N. Model Based Training, Detection and Pose Estimation of Texture-Less 3d Objects in Heavily Cluttered Scenes. In Proceedings of the ACCV'12, Berlin, Heidelberg, 2012; p. 548–562. https://doi.org/10.1007/978-3-642-37331-2_42.
23. Garrido-Jurado, S.; Muñoz Salinas, R.; Madrid-Cuevas, F.; Marín-Jiménez, M. Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion. *Pattern Recogn.* **2014**, *47*, 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005.
24. Wang, P.; Jung, H.; Li, Y.; Shen, S.; Srikanth, R.P.; Garattoni, L.; Meier, S.; Navab, N.; Busam, B. PhoCaL: A Multi-Modal Dataset for Category-Level Object Pose Estimation With Photometrically Challenging Objects. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 21222–21231.
25. Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M.J.; Laptev, I.; Schmid, C. Learning from Synthetic Humans. In Proceedings of the CVPR, 2017.
26. Hodaň, T.; Haluza, P.; Obdržálek, Š.; Matas, J.; Lourakis, M.; Zabulis, X. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)* **2017**.
27. Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; Guibas, L.J. Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

 doi:10.20944/preprints202602.0443.v1

28. Brachmann, E. 6D Object Pose Estimation using 3D Object Coordinates [Data], 2020. https://doi.org/10.11588/data/V4MUMX.
29. Chen, B.; Chin, T.J.; Klimavicius, M. Occlusion-Robust Object Pose Estimation with Holistic Representation. In Proceedings of the WACV, 2022.
30. Bukschat, Y.; Vetter, M. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach, 2020, [arXiv:cs.CV/2011.04307].
31. Wang, G.; Manhardt, F.; Liu, X.; Ji, X.; Tombari, F. Occlusion-Aware Self-Supervised Monocular 6D Object Pose Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, pp. 1–1. https://doi.org/10.1109/tpami.2021.3136301.
32. Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. *CoRR* **2018**, *abs/1809.10790*, [1809.10790].
33. Li, S.; Xu, C.; Xie, M. A Robust O(n) Solution to the Perspective-n-Point Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2012**, *34*, 1444–1450. https://doi.org/10.1109/TPAMI.2012.41.
34. Park, K.; Patten, T.; Vincze, M. Pix2Pose: Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In Proceedings of the The IEEE International Conference on Computer Vision (ICCV), Oct 2019.
35. Deng, X.; Xiang, Y.; Mousavian, A.; Eppner, C.; Bretl, T.; Fox, D. Self-supervised 6D Object Pose Estimation for Robot Manipulation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3665–3671. https://doi.org/10.1109/ICRA40945.2020.9196714.
36. Iwase, S.; Liu, X.; Khirodkar, R.; Yokota, R.; Kitani, K.M. RePOSE: Fast 6D Object Pose Refinement via Deep Texture Rendering. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 3303–3312.
37. Di, Y.; Manhardt, F.; Wang, G.; Ji, X.; Navab, N.; Tombari, F. SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 12396–12405.
38. He, Y.; al., e. FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021.
39. Nguyen, V.N.; Hu, Y.; Xiao, Y.; Salzmann, M.; Lepetit, V. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6771–6780.
40. Cai, D.; Heikkilä, J.; Rahtu, E. OVE6D: Object Viewpoint Encoding for Depth-Based 6D Object Pose Estimation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 6803–6813.
41. Gu, Q.; Okorn, B.; Held, D. OSSID: Online Self-Supervised Instance Detection by (And For) Pose Estimation. *IEEE Robotics and Automation Letters* **2022**, *7*, 3022–3029. https://doi.org/10.1109/LRA.2022.3145488.
42. Shugurov, I.; Li, F.; Busam, B.; Ilic, S. OSOP: A Multi-Stage One Shot Object Pose Estimation Framework. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 6835–6844.
43. Liu, Y.; Wen, Y.; Peng, S.; Lin, C.; Long, X.; Komura, T.; Wang, W. Gen6D: Generalizable Model-Free 6-DoF Object Pose Estimation from RGB Images. In Proceedings of the ECCV, 2022.
44. Sun, J.; Wang, Z.; Zhang, S.; He, X.; Zhao, H.; Zhang, G.; Zhou, X. OnePose: One-Shot Object Pose Estimation without CAD Models. *CVPR* **2022**.
45. Chen, H.; Manhardt, F.; Navab, N.; Busam, B. TexPose: Neural Texture Learning for Self-Supervised 6D Object Pose Estimation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023, pp. 4841–4852.
46. Stoiber, M.; Elsayed, M.; Reichert, A.E.; Steidle, F.; Lee, D.; Triebel, R. Fusing Visual Appearance and Geometry for Multi-modality 6DoF Object Tracking. *arXiv preprint arXiv:2302.11458* **2023**.
47. Stoiber, M.; Sundermeyer, M.; Triebel, R. Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6855–6865.
48. Gard, N.; Hilsmann, A.; Eisert, P. CASAPose: Class-Adaptive and Semantic-Aware Multi-Object Pose Estimation. In Proceedings of the 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022. BMVA Press, 2022.
49. Zuo, Y.; Qiu, W.; Xie, L.; Zhong, F.; Wang, Y.; Yuille, A.L. Craves: Controlling robotic arm with a vision-based economic system. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4214–4223.

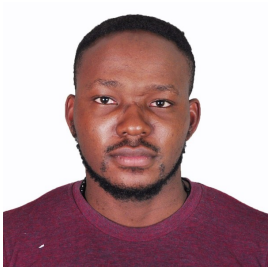
 doi:10.20944/preprints202602.0443.v1

50. Karashchuk, P.; Rupp, K.L.; Dickinson, E.S.; Walling-Bell, S.; Sanders, E.; Azim, E.; Brunton, B.W.; Tuthill, J.C. Anipose: a toolkit for robust markerless 3D pose estimation. *Cell reports* **2021**, *36*, 109730.
51. Labbé, Y.; Carpentier, J.; Aubry, M.; Sivic, J. Single-view robot pose and joint angle estimation via render & compare. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1654–1663.
52. Noguchi, A.; Iqbal, U.; Tremblay, J.; Harada, T.; Gallo, O. Watch It Move: Unsupervised Discovery of 3D Joints for Re-Posing of Articulated Objects. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 3677–3687.
53. Simoni, A.; Pini, S.; Borghi, G.; Vezzani, R. Semi-Perspective Decoupled Heatmaps for 3D Robot Pose Estimation From Depth Maps. *IEEE Robotics and Automation Letters* **2022**, *7*, 11569–11576. https://doi.org/10.1109/LRA.2022.3193225.
54. Tian, Y.; Zhang, J.; Yin, Z.; Dong, H. Robot Structure Prior Guided Temporal Attention for Camera-to-Robot Pose Estimation From Image Sequence. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023, pp. 8917–8926.
55. Wu, Y.; Chen, Z.; Liu, S.; Ren, Z.; Wang, S. CASA: Category-agnostic Skeletal Animal Reconstruction. In Proceedings of the Advances in Neural Information Processing Systems; Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; Oh, A., Eds. Curran Associates, Inc., 2022, Vol. 35, pp. 28559–28574.
56. ISO. Road vehicles – Functional safety, 2011.
57. DALLEL, M.; HAVARD, V.; BAUDRY, D.; SAVATIER, X. InHARD - Industrial Human Action Recognition Dataset in the Context of Industrial Collaborative Robotics. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), 2020, pp. 1–6.
58. Park, K.; Mousavian, A.; Xiang, Y.; Fox, D. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10710–10719.

Short Biography of Authors

Maximillian Panoff Maximillian 'Max' Panoff (Student Member, IEEE) is a fourth-year Ph.D. student in the Department of Electrical and Computer Engineering. he earned his B.E. with a Concentration in Robotics and a Minor in Social Sciences from Stevens Institute of Technology in Hoboken, NJ. During his time at the University of Florida where he studies human collaborative robotics, he was awarded the Graduate School Preeminence Award.

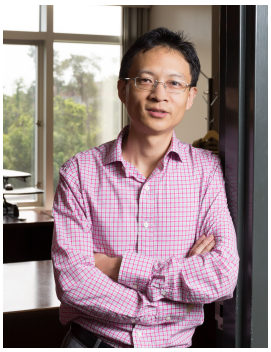
Antonio Hendricks Antonio Hendricks (Student Member, IEEE) joined the Smart Systems Lab as a Ph.D. Student and Graduate Research Assistant at the University of Florida in the ECE Department in the Fall 2022 semester. He completed his Bachelor in Computer Engineering with a focus on Machine Intelligence and Autonomous Robotics Systems at Florida Polytechnic University in Lakeland Florida. He plans to focus on safe autonomy and advanced robotic intelligence for surgery, utilizing modern methods of deep learning and computer vision.



Peter Forcha Peter Forcha is currently pursuing a Ph.D. in the Department of Electrical and Computer Engineering at the University of Florida. He obtained his bachelor's degree in Electrical and Electronics Engineering at the University of Buea (Cameroon) and completed his Master of Engineering in Power Systems at Makerere University (Uganda). Peter's research is focused on building hardware accelerators for machine learning along other machine learning applications, such as computer vision and natural language processing.

 doi:10.20944/preprints202602.0443.v1

Minchul Jung Minchul Jung is a first-year Graduate student at the University of Florida. He works in the Electrical and Computer Engineering department and conducts research with the Smart Systems Lab, led by Dr. Christophe Bobda. His research interests include Human-robot interaction and computer vision. He holds a bachelors degree in Aerospace Engineering from Korea Airforce Academy.



Shuo Wang Shuo Wang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Virginia Tech, Blacksburg, VA, USA, in 2005.,He is currently a Full Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. He has authored or coauthored more than 200 IEEE journal and conference papers and holds around 30 pending/issued U.S./international patents.,Dr. Wang was the recipient of the Best Transaction Paper Award from the IEEE Power Electronics Society in 2006 and two William M. Portnoy Awards for the papers published in the IEEE Industry Applications Society in 2004 and 2012, respectively. He received the Distinguished Paper Award from IEEE Symposium on Security and Privacy 2022. In 2012, he was also the recipient of the prestigious National Science Foundation CAREER Award. He is an Associate Editor of the IEEE Transactions on Industry Applications and IEEE Transactions on Electromagnetic Compatibility. He was a Technical Program Co-Chair for the IEEE 2014 International Electric Vehicle Conference.

Professor Bobda Christophe Bobda (Member, IEEE) received a Licence in mathematics from the University of Yaounde, Cameroon, in 1992, a diploma of computer science and a Ph.D. degree (with honors) in computer science from the University of Paderborn in Germany in 1999 and 2003 (In the chair of Prof. Franz J. Rammig) respectively. In June 2003 he joined the department of computer science at the University of Erlangen-Nuremberg in Germany as Post doc, under the direction of Prof Jürgen Teich. Dr. Bobda received the best dissertation award 2003 from the University of Paderborn for his work on synthesis of reconfigurable systems using temporal partitioning and temporal placement. In 2005 Dr. Bobda was appointed assistant professor at the University of Kaiserslautern. There he set the chair for Self-Organizing Embedded Systems that he led until October 2007. From 2007 to 2010 Dr. Bobda was a Professor at the University of Potsdam and leader of The working Group Computer Engineering.