Domain-Translated 3D Object Pose Estimation

Christos Papaioannidis, Vasileios Mygdalis and Ioannis Pitas, *Fellow, IEEE*

***Abstract*—Synthetic 3D object models have been proven crucial in object pose estimation, as they are utilized to generate a huge number of accurately annotated data. The object pose estimation problem is usually solved for images originating from the real data domain by employing synthetic images for training data enrichment, without fully exploiting the fact that synthetic and real images may have different data distributions. In this work, we argue that 3D object pose estimation problem is easier to solve for images originating from the synthetic domain, rather than the real data domain. To this end, we propose a 3D object pose estimation framework consisting of a two-step process, where a novel pose-oriented image-to-image translation step is first employed to translate noisy real images to clean synthetic ones and then, a 3D object pose estimation method is applied on the translated synthetic images to finally predict the 3D object poses. A novel pose-oriented objective function is employed for training the image-to-image translation network, which enforces that pose-related object image characteristics are preserved in the translated images. As a result, the pose estimation network does not require real data for training purposes. Experimental evaluation has shown that the proposed framework greatly improves the 3D object pose estimation performance, when compared to state-of-the-art methods.**

***Index Terms*—3D object pose estimation, image-to-image translation, domain-translated, synthetic data.**

I. Introduction

OBJECT pose estimation is a very important computer vision task in autonomous driving, robotic manipulation and augmented reality applications that attracted significant research focus over the past few years. Although object pose estimation has recently been heavily researched, it remains a very challenging task, mostly due to strong variations on the object appearance imposed by illumination and scale changes that restrict the practical use cases of most pose estimation methods, limiting their applicability only to restricted, controlled environments.

Object pose estimation involves estimating the 3D orientation and 3D translation of an object of interest, relative to a camera coordinate system $(\mathbf{O}_c, X_c, Y_c, Z_c)$, commonly referred to as the 6D object pose estimation problem. The estimation of 3D object translation is typically approached by localizing the object on the 2D image plane, using e.g., an object detector, and then estimating the 3D translation using the camera intrinsic parameters. Estimating the 3D object orientation (3D object pose estimation) is of particular interest that is usually solved separately [1]–[5] from the 3D translation problem. This work focuses on estimating the 3D object poses of already localized objects, where the rotation between the object $(\mathbf{O}_o, X_o, Y_o, Z_o)$ and the camera coordinate systems needs to be estimated, e.g., in a form of a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ or a unit quaternion $\mathbf{q} \in \mathbb{R}^4, \|\mathbf{q}\|_2 = 1$.

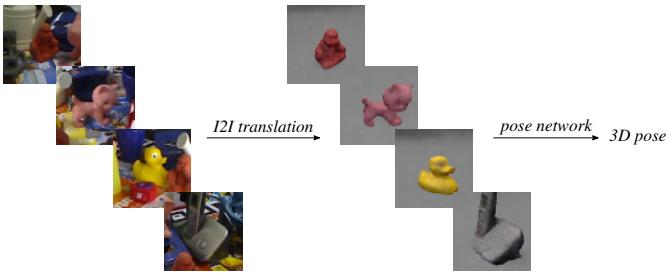


Fig. 1. Methodology of the proposed 3D object pose estimation framework (translating noisy real images to clean synthetic ones and, then, estimating the corresponding 3D object poses using the translated images in a 3D object pose estimation network).

Many recent object pose estimation methods [1]–[11] exploited the power of Convolutional Neural Networks (CNNs) [12] to achieve accurate pose estimation results. However, it is well known that CNNs usually require a huge amount of training data. Synthetic data have been proven to be invaluable to this end, especially for object pose estimation methods, where annotating an object image with its ground truth 3D pose can be an extremely difficult, inaccurate and time-consuming process. The simplest approach is to exploit synthetic data as an additional data source, assuming that real and synthetic images belong to the same image domain [1]–[4], [6], [9]–[11], [13]–[16]. The disadvantage of such approaches is related to the fact that real and synthetic images actually originate from two separate image domains, which is not taken into account. On the other hand, object pose estimation methods that consider this property typically separate the real and synthetic images in two domains, aiming to either generate "fake" data to train the pose estimation network [17], [18], or learn domain-invariant features [19]–[23]. To this end, image-to-image translation (I2I) [24]–[26] can be employed for learning domain-independent semantic object representations, in order to change a particular aspect of an image to another one [27]. Such approaches have found successful application in semantic image segmentation [28]. However, as the ultimate goal is to apply the trained object pose estimation models in unconstrained environments (i.e. for drone cinematography), methods for agnostic feature learning fail to successfully generalize to real images, whose domain distribution typically diverges from the one of the training data.

In this work, we argue that assuming a single domain for real and synthetic images, or trying to learn domain

Christos Papaioannidis, Vasileios Mygdalis and Ioannis Pitas are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. e-mail: cpapaionn@csd.auth.gr, mygdalisv@csd.auth.gr, pitas@aiia.csd.auth.gr

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 871479 (AERIAL-CORE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

TABLE I
TOTAL LOSS FUNCTIONS USED IN THE METHODS OF [1]–[4].

[1]	$\mathcal{L} = \sum \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 + \sum max\big(0, 1 - \frac{\Delta_-}{\Delta_+ + \varepsilon}\big) + \lambda \|\mathbf{w}\|_2^2$
[2]	$\mathcal{L} = \sum \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 + \sum max\big(0, 1 - \frac{\Delta_-}{\Delta_+ + \varepsilon_d}\big) + \lambda \|\mathbf{w}\|_2^2$, $\varepsilon_d = \begin{Bmatrix} 2\arccos(\lvert\mathbf{q}_i^T \mathbf{q}_j\rvert) & if\ c_i{=}c_j, \\ n & else, for\ n{>}\pi \end{Bmatrix}$
[3]	$\mathcal{L} = \sum \{\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 - \|\mathbf{p}_i - \mathbf{p}_j\|_2^2\}^2 + \sum \frac{\Delta_+}{\Delta_- + \varepsilon} + \sum \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$
[4]	$\mathcal{L} = \sum \{\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 - 2\arccos(\lvert\mathbf{q}_i^T \mathbf{q}_j\rvert)\}^2 + \sum \frac{\Delta_+}{\Delta_- + \varepsilon} + \sum \|\mathbf{q}_i - \hat{\mathbf{q}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$

agnostic image features is a sub-optimal approach for 3D object pose estimation. We consider that 3D object pose estimation is easier to solve in the synthetic image domain, rather than the real image domain, as synthetic data do not suffer from excessive noise that is typically encountered in real object images. The final 3D object poses can thereby be best estimated from clean, synthetic images. To this end, we propose a 3D object pose estimation framework that incorporates a pose-oriented I2I step that translates real object images to synthetic ones and a 3D object pose estimation step that acts on the translated images, as depicted in Fig. 1. The pose-oriented I2I network of the proposed framework is tasked to not only generate synthetic images that are clean from background clutter, blurring and illumination variations that affect object appearance, but also to preserve the 3D pose-related object image characteristics that enable the pose estimation network to perform more accurate 3D object pose estimation. In order to accomplish both these goals, a novel pose-oriented objective function is proposed to train the pose-oriented I2I network. Finally, note that applying I2I prior to 3D object pose estimation enables the 3D object pose estimation network to be trained solely on synthetic data.

In summary, this paper offers the following novel contributions:

- introduction of a novel 3D object pose estimation framework, which consists of a pose-oriented image-to-image translation step and a 3D object pose estimation step;
- a novel pose-oriented loss function is proposed to train the image-to-image translation network to generate clean synthetic images where the pose-related object image characteristics are preserved;
- significant improvement in 3D object pose estimation accuracy, compared to state-of-the-art (SoA) methods.

The rest of this paper is organized as follows. Previous object pose estimation methods are reviewed in Section II. The proposed pose-oriented I2I network, as well as the full 3D object pose estimation framework are described in Section III. The experimental setup and the extensive evaluation of the proposed method compared to SoA 3D object pose estimation methods can be found in Section IV. Finally, conclusions are drawn in Section V.

II. RELATED WORK

In this section, previous work related to 3D object pose estimation is summarized. Two main methodological approximations are discussed, depending on the data used during training. Finally, feature learning object pose estimation methods are reviewed as they are closely related to our work.

The first methodological family represents object pose estimation methods that either employ solely real images for CNN training [7], [8], [29]–[34], or use synthetic images to simply augment the training dataset, assuming that real and synthetic images belong to the same image domain [1]–[4], [6], [9]–[11], [13]–[16]. More specifically, real images were employed to train a CNN to either directly predict 3D object poses by regression [30], [32], or to classify the object image to a set of predefined 3D orientation classes that were defined by an appropriate quantization of the continuous 3D pose space [29], [31]. In a similar fashion, a CNN was trained to perform 3D object pose regression but this time by augmenting the training dataset with rendered synthetic images [13], [15]. However, these methods require pre-trained deep CNNs [13], [30] or a separate 3D pose estimation network for each object of interest [15]. As directly estimating 3D object poses is a difficult task due to the non-linearity of the 3D object pose space, the 3D object pose estimation problem was approached as a nearest neighbor (NN) search one in [1]–[5], [16]. By using a mixture of real and synthetic images to train a lightweight CNN as a feature extractor and matching the learned object image feature vector with a set of 3D orientation class representatives (vectors), these methods managed to considerably increase the 3D object pose estimation performance. However, the fact that real and synthetic images originate from two separate image domains was also not considered. Furthermore, object detection [35], [36] and semantic image segmentation [37] network architectures were extended to estimate the 6D poses of multiple objects in an image [6], [9]–[11], [33], [34]. At first, all objects are localized on the image by regressing their 2D bounding boxes or a set of predefined 2D control points. Then, the 2D object detections are utilized by a second, 6D pose estimation step to predict the final 6D poses. Similarly, an autoencoder [38] architecture was employed in [8] to estimate 3D coordinates and expected errors for each pixel in the input images. These pixel-wise predictions are then used to compute 6D poses using a PnP [39] algorithm. Also in all these cases, synthetic data are either ignored, or are simply used to enrich the training dataset.

The second methodological family consists of methods which consider the fact that real and synthetic images originate from two separate image domains [17]–[23]. More specifically, [22], [23] exploited the available synthetic 3D mesh models to create a training set which consisted solely of synthetic object images for training their 6D object pose estimation

networks. However, in order for these networks to be able to generalize to real object images, advanced image augmentation techniques were required. In another direction, [19] aimed to learn domain-invariant features for the classification and 3D object pose estimation tasks by partitioning the image features in two subspaces, one that is private to each image domain and the other that is shared across domains. Subsequently, a neural network was employed in [20], in order to align the real image features distribution with the one of the synthetic image features for 3D object pose estimation. Finally, [17], [18] utilized Generative Adversarial Networks (GANs) [40] to generate "fake" real object images, which are then used along synthetic object images to train the 3D object pose estimation framework. It has to be noted that the proposed 3D object pose estimation framework highly differs from all these approaches, as it utilizes I2I to transfer real object images to the synthetic image domain as a preceding step and performs 3D object pose estimation on the translated synthetic object images.

As the 3D object pose estimation step of the proposed framework is closely related to [1]–[4], a more detailed analysis of these methods follows. In feature learning methods [1]–[4], the objective is to learn lower-dimensionality discriminant 3D object pose features to be used in 3D object pose inference. More specifically, the extracted 3D object pose feature vector is matched with a precomputed feature vector database via a NN search in the feature space, returning the corresponding ground truth poses as the 3D object pose estimations. One advantage of these methods is that they are scalable to the number of objects [2]. Also, they can generalize to unseen objects, without the need to train additional pose estimation models [4]. Finally, apart from 3D object pose estimation, these methods also enable object classification, by returning the matched database object labels, besides their 3D pose estimates. In order to learn such pose features, a lightweight CNN was trained as a feature extractor in [1], using the following loss function:

$$\mathcal{L} = \mathcal{L}_d + \lambda \|\mathbf{w}\|_2^2, \tag{1}$$

where $\mathcal{L}_d$ is related to feature learning, λ is a regularization parameter and $\|\mathbf{w}\|_2$ is the L_2-norm of the network parameter vector. $\mathcal{L}_d$ consists of a pairwise [41] $\mathcal{L}_{pairs}$ and a triplet [42] $\mathcal{L}_{triplets}$ loss function, $\mathcal{L}_d = \mathcal{L}_{pairs} + \mathcal{L}_{triplets}$, in order to learn object image features, which allow both 3D object label and pose estimation. $\mathcal{L}_{pairs}$ forces the distances between features of object images with similar poses to be small, while $\mathcal{L}_{triplets}$ is responsible for discriminating features of object images of different object classes. Therefore, by minimizing $\mathcal{L}_d$ during CNN training, object classes and 3D object pose sub-clusters are formed in the feature space, enabling 3D object pose estimation and object classification through NN search in this feature space. In order to learn more discriminating 3D object pose features, a dynamic margin has been added in the triplet loss function $\mathcal{L}_{triplets}$ [2]:

$$\varepsilon_d = \begin{cases} 2\arccos(|\mathbf{q}_i^T \mathbf{q}_j|) & if\ c_i = c_j, \\ n & \text{else, for } n > \pi, \end{cases} \tag{2}$$

where $\mathbf{q}_i$, $\mathbf{q}_j$ and c_i, c_j are the corresponding ground truth 3D training sample pose and object identity labels s_i, s_j, respectively. More recent methods [3]–[5] showed that feature learning can be enhanced by adding an extra 3D pose regression term to the total loss function (1). Therefore, the total loss function becomes of the following form:

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_{reg} + \lambda \|\mathbf{w}\|_2^2, \tag{3}$$

where $\mathcal{L}_{reg}$ is the 3D object pose regression term, which is a standard RMS error over all samples. Finally, representing 3D object orientations with unit quaternions has been proven to further increase the performance of feature learning methods [4]. By using the inverse cosine quaternion distance [43] in the pairwise loss function $\mathcal{L}_{pairs}$ and quaternion regression as the regression term $\mathcal{L}_{reg}$, the CNN yielded more robust pose features, which improved the 3D object pose estimation performance as well as the generalization ability of the trained model to unknown objects [4]. An overview of the total loss functions used in [1]–[4] is presented in Table I.

III. Domain-Translated (DT) 3D Object Pose Estimation

Synthetic 3D object models improved 3D object pose estimation performance, as they can create a huge number of accurately annotated synthetic object images. However, synthetic images require special attention, as their appearance greatly differs from the one of real images, thus often reducing CNN generalization ability. Taken this into account, the proposed novel method treats real and synthetic images as data originating from different image domains $\mathcal{X}$, $\mathcal{Y} \subseteq \mathbb{R}^D$ for the real and synthetic images, respectively. We propose that 3D object pose estimation benefits from translating a real image $\mathbf{x} \in \mathcal{X}$ to a synthetic image $\mathbf{y} \in \mathcal{Y}$, by clearing image noise that affect object appearance, while retaining 3D object pose information. The final 3D object poses are then estimated from the clean translated images. Therefore, the proposed 3D object pose estimation framework consists of an image-to-image translation step and a 3D object pose estimation step. The two components of the proposed framework are described in the following subsections.

A. Pose-oriented image-to-image translation

Given a pair of real and synthetic training samples $\{\mathbf{x}_i, \mathbf{y}_i\}$, $i = 1 \dots N$, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$, image-to-image translation methods [24] aim to learn a mapping function G from the real to the synthetic image domain, $G : \mathcal{X} \mapsto \mathcal{Y}$. In this work, G is represented by the generator of a conditional GAN [44]. Conditional GANs consist of two competing networks, the so-called generator G and the discriminator D. Given samples originating from the source domain, the generator G aims to produce outputs that mimic target domain samples and cannot be distinguished by the discriminator D, which is adversarially trained to detect the generator's "fake" outputs. In our case, the objective of the generator G is to learn the underlying mapping from the real object image domain $\mathcal{X}$ (source domain) to the synthetic object image domain $\mathcal{Y}$ (target domain), while the objective of the discriminator D is to distinguish samples $G(\mathbf{x}_i)$ produced by the generator from target domain samples $\mathbf{y}_i \in \mathcal{Y}$ (synthetic

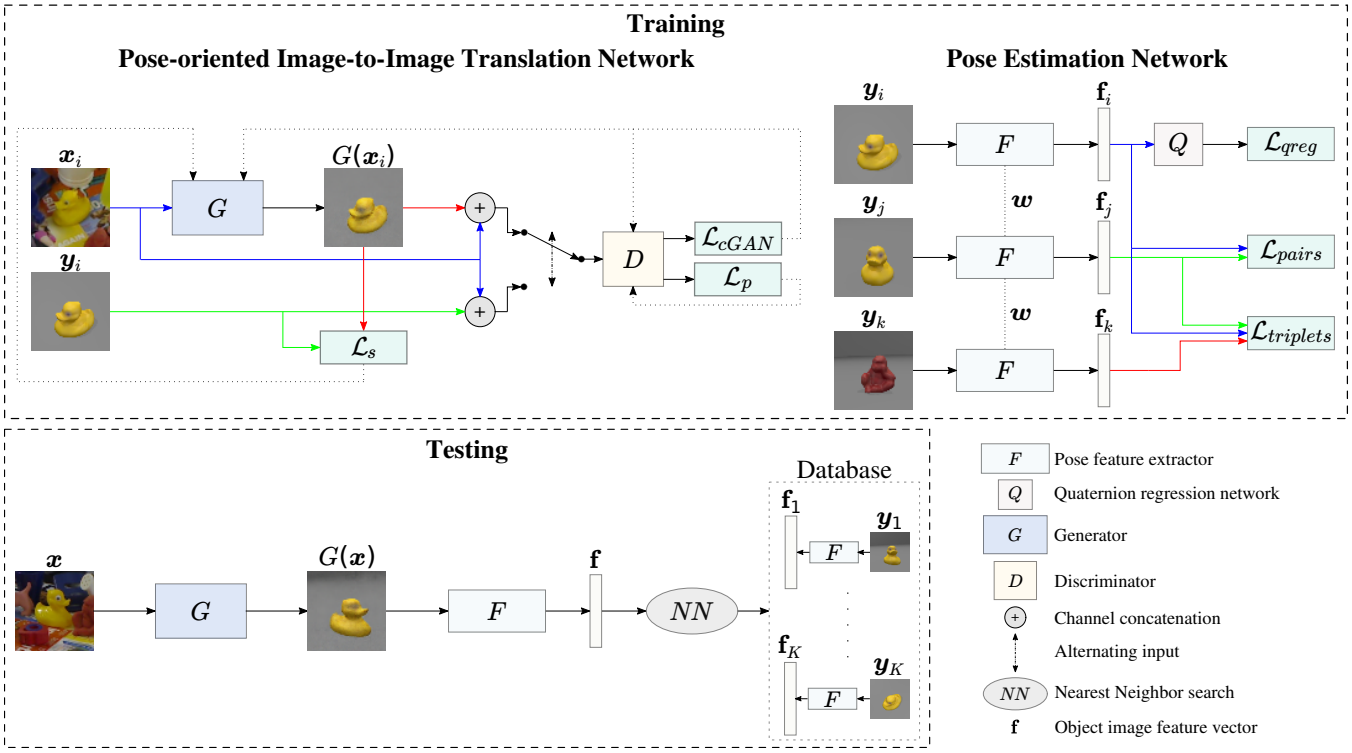


Fig. 2. Training and testing of the proposed 3D object pose estimation framework. The image-to-image translation and the pose estimation networks are trained separately. During inference, the trained generator G translates real object images to synthetic ones, which are then provided as input to the trained pose feature extractor F. Finally, the extracted image features $\mathbf{f}$ are matched with a set of precalculated database image features via NN search.

image domain). As in typical conditional GANs, both G and D are trained in a supervised manner via a minimax game. Specifically, G parameters are updated so that the objective function $\mathcal{L}_{cGAN}(G,D)$ is minimized while D on the other hand maximizes it, i.e. $\min_G \max_D \mathcal{L}_{cGAN}(G,D)$, where $\mathcal{L}_{cGAN}(G,D)$ is given by [24]:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x}}[\log(1 - D(\mathbf{x}, G(\mathbf{x})))]. \quad (4)$$

In order to achieve best 3D object pose estimation performance in the proposed framework, a similarity loss function is also required to keep the translated images $G(\mathbf{x})$ as similar as possible to synthetic domain images $\mathbf{y}$. In alignment with previous methods [24], [45], we also train the generator to not only fool the discriminator, but also to generate images that are "close" to the corresponding synthetic domain images in a $L2$ distance sense. Therefore, the employed similarity loss function is defined as:

$$\mathcal{L}_s(G) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}[\|\mathbf{y} - G(\mathbf{x})\|_2^2]. \quad (5)$$

The use of training labels has also been proven [46] to assist GAN training and to further improve the quality of the generated images. While using object class labels is a common practice [47]–[49], we propose utilizing the real 3D object pose quaternions to the conditional GAN training process. More specifically, apart from distinguishing between synthetic and translated images, the discriminator D is also tasked to predict the depicted object 3D pose in the form of a unit quaternion $\mathbf{q}$. To this end, an additional quaternion regression loss function [4] is also used for discriminator training:

$$\mathcal{L}_p(D) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{q} - \hat{\mathbf{q}}\|_2^2, \quad (6)$$

where $\mathbf{q}$ and $\hat{\mathbf{q}}$ are the ground truth and the predicted 3D object pose quaternions, respectively. Essentially, the quaternion regression loss function $\mathcal{L}_p(D)$ renders the discriminator more sensitive to 3D object pose variation. As a result, the generator is forced to generate synthetic images of higher quality in order to fool the discriminator, while retaining the 2D object image characteristics that allow better 3D object pose estimation on the translated images.

The final objective function used to train the proposed pose-oriented I2I network is thereby defined as follows:

$$\mathcal{L}_{i2i} = \min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda_s \mathcal{L}_s(G) + \lambda_p \mathcal{L}_p(D), \quad (7)$$

where λ_s, λ_p are hyper-parameters used to control the relative importance of reconstruction and quaternion regression losses, respectively, compared to the adversarial loss. By using (7) in the conditional GAN training framework, the generator G learns not only to accurately translate real object images to the corresponding synthetic ones, but to also to retain the specific object characteristics that enable 3D object pose estimation by the pose estimation network using the translated images as inputs. Examples of real and synthetic RGB-D images along with the translated images can be seen in Fig. 3. It can be easily seen that the proposed method removes image

background, while retaining image appearance characteristics that are crucial for 3D object pose estimation.

B. Proposed 3D object pose estimation framework

The overall proposed 3D object pose estimation framework is illustrated in Fig. 2. The 3D pose-oriented I2I and 3D object pose estimation networks are trained separately. In order to train the proposed I2I network, a specially designed training set $\mathcal{S}_{i2i}$ is required, which consists of samples $s_{i2i} = \{\mathbf{x}_i, \mathbf{y}_i, \mathbf{q}_i\}$, where $\mathbf{x}_i \in \mathcal{X}$ is a real image depicting an already localized object under a specific pose, $\mathbf{y}_i \in \mathcal{Y}$ is the paired synthetic image of the same object under the same pose and $\mathbf{q}_i$ is the ground truth 3D object pose that corresponds to $\mathbf{x}_i$. The real object image $\mathbf{x}_i$ is given as input to the I2I network, while the paired synthetic image $\mathbf{y}_i$ acts as the target image for training G. The ground truth 3D object pose $\mathbf{q}_i$ is used as label to train the discriminator network D for the extra pose estimation task. Both G and D are then trained using $\mathcal{S}_{i2i}$ under the conditional GAN training framework described in III-A.

The 3D object pose estimation network consists of a lightweight CNN F, and a quaternion regression network Q. F is trained as a feature extractor by combining a pairwise, a triplet and a quaternion regression loss function, while Q is trained to further utilize the learned features to regress 3D object poses in unit quaternions representation. Thus, the multi-objective loss function to be minimized is given by [4]:

$$\mathcal{L}_{pose3d} = \sum \{\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 - 2\arccos(|\mathbf{q}_i^T \mathbf{q}_j|)\}^2 + \sum \frac{\|\mathbf{f}_i - \mathbf{f}_j\|_2}{\|\mathbf{f}_i - \mathbf{f}_k\|_2 + \varepsilon} + \sum \|\mathbf{q}_i - \hat{\mathbf{q}}_i\|_2^2, \quad (8)$$

where $\mathbf{f}_i$ denotes the object image features extracted by the CNN, $|\mathbf{q}_i^T \mathbf{q}_j|$ is the absolute value of the inner product between $\mathbf{q}_i$, $\mathbf{q}_j$ and ε is a small regularizing constant. For a more detailed explanation about the pair and triplet samples used in the pairwise and triplet loss, respectively, the interested reader can refer to [4].

During testing, the trained I2I and pose estimation networks are utilized in a unified pipeline to estimate 3D object pose quaternions $\hat{\mathbf{q}}$. First, a real image $\mathbf{x}$ is translated to a synthetic one $G(\mathbf{x})$ using the I2I network and then, the translated image is provided to the pose estimation network, which outputs the final 3D pose prediction. Let $\mathbf{f} \in \mathcal{F} \subset \mathbb{R}^d$ be object image features obtained from the trained pose feature extraction network F. Also, let samples $s_{db} = \{\mathbf{y}_i, c_i, \mathbf{q}_i\}$, $i = 1 \dots K$, comprise a database set $\mathcal{S}_{db}$, where $\mathbf{y}_i \in \mathcal{Y}$ are synthetic object images and $c_i \in \mathcal{C} = \{c_1, \dots, c_L\}$, $\mathbf{q}_i$ are the corresponding object identity labels and 3D object poses, respectively. Therefore, the final 3D object pose estimations are obtained by matching the calculated image features $\mathbf{f} = F(G(\mathbf{x}))$ with a set of database features $\mathbf{f}_i = F(\mathbf{y}_i) \in \mathcal{F}$ via a NN search in the feature space. Practically, the database image features $\mathbf{f}_i$ are computed offline using the trained pose feature extractor F and are stored in a look-up table which is available during testing. Note that $\mathcal{S}_{db}$ is only used during inference, acting as a database for matching test real object images with database ones via a NN search algorithm.

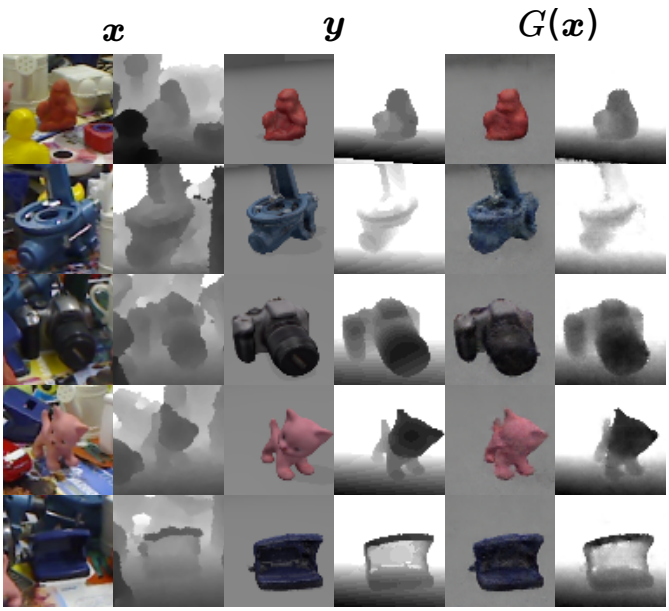


Fig. 3. Examples of real ($\mathbf{x}$), synthetic ($\mathbf{y}$) and translated ($G(\mathbf{x})$) RGB-D images obtained from the proposed pose-oriented I2I network.

Another advantage of the proposed framework is that since real images are first translated to synthetic ones and then are given to the 3D object pose estimation network to obtain the final estimations, it allows the pose estimation network to be trained using a training dataset that may consist of either a mixture of real and synthetic images, or purely synthetic images. In addition, the modular design of the proposed testing pipeline enables replacing the selected 3D object pose estimation method, with any state-of-the-art object pose estimation method. Finally, it has to be noted that the proposed 3D object pose estimation framework does not insist on achieving optimal visual quality for the image-to-image translation task, as long as the specific pose characteristics are preserved in the translated images.

IV. Experimental Results

In this section, we describe the experiments conducted in order to evaluate the performance of the proposed framework. The employed network architectures and relevant details about the implementation are provided in Subsection IV-A. Our experimental protocol and the detailed comparison between the proposed 3D object pose estimation framework and SoA object pose estimation methods are described in Subsection IV-B. Finally, in Subsection IV-C, an ablation study is conducted to highlight the importance of each component in the proposed framework.

A. Implementation details

The generator network G and the discriminator D are trained in an adversarial fashion in the conditional GAN framework. An encoder-decoder network architecture [38] is employed for G which is similar to the one used in [23], with some key variations. More specifically, the convolutional layer kernel size was reduced from 5×5 to 3×3, both in the encoder

TABLE II
3D OBJECT POSE ESTIMATION AND OBJECT CLASSIFICATION ACCURACY IN RGB AND RGB-D SETTING.

	Angular threshold t								
	5°	**10°**	**15°**	**20°**	**30°**	**40°**	**45°**	**Mean (Median) ± Std**	**Object classification**
RGB									
$3DPOD$ [1]	36.47%	64.11%	77.32%	83.87%	89.37%	91.78%	92.71%	16.38°(8.10°) ± 27.24°	96.11%
FM [20]	32.63%	66.70%	86.99%	95.49%	99.57%	**99.93%**	**99.95%**	8.42°(7.35°) ± 6.15°	86.55%
$PGFL$ [3]	37.19%	80.30%	92.73%	96.26%	98.49%	99.14%	99.26%	7.61°(6.12°) ± 8.57°	98.80%
QL [4]	40.15%	79.42%	93.66%	**97.77%**	**99.63%**	**99.93%**	**99.95%**	6.87°(5.91°) ± 5.08°	98.80%
$PixelDA$ [17]	35.72%	74.83%	91.13%	96.96%	99.55%	99.88%	99.89%	7.66°(6.56°) ± 5.94°	98.71%
$PixelDA_{f+r}$ [17] *	21.14%	54.26%	79.60%	91.65%	98.56%	99.54%	99.61%	10.51°(9.35°) ± 7.19°	98.49%
$PixelDA_{f+s+r}$ [17], [28] *	37.41%	77.78%	93.06%	97.09%	99.49%	99.86%	99.90%	7.28°(6.24°) ± 5.31°	98.87%
DT *(ours)*	51.08%	83.97%	**93.78%**	97.29%	98.89%	99.43%	99.53%	**6.45**°(4.91°) ± 7.06°	**99.01%**
DT_{r+s} *(ours)* †	**52.18%**	**84.43%**	93.47%	96.50%	98.33%	98.88%	99.08%	6.66°(**4.80**°) ± 8.28°	98.33%
DT_r *(ours)* ‡	47.18%	78.08%	89.44%	94.06%	97.29%	98.46%	98.71%	7.87°(5.27°) ± 10.27°	98.80%
RGB-D									
$PEDM$ [2] ⋆	-	60.00%	-	93.20%	-	98.00%	-	-	99.30%
$3DPOD$ [1]	40.15%	72.72%	86.02%	91.76%	95.42%	96.90%	97.34%	12.75°(7.06°) ± 24.61°	98.94%
$PGFL$ [3]	41.28%	83.07%	93.98%	97.43%	99.11%	99.52%	99.60%	6.89°(5.79°) ± 6.29°	**99.64%**
QL [4]	41.37%	82.02%	95.32%	**98.49%**	**99.72%**	**99.92%**	**99.94%**	6.64°(5.78°) ± 5.14°	99.50%
$PixelDA$ [17]	29.56%	66.40%	86.90%	94.59%	98.86%	99.66%	99.82%	8.80°(7.63°) ± 6.39°	98.13%
$PixelDA_{f+r}$ [17] *	21.09%	55.77%	80.39%	91.43%	97.72%	99.10%	99.37%	10.64°(9.13°) ± 8.34°	95.41%
$PixelDA_{f+s+r}$ [17], [28] *	43.16%	84.80%	95.41%	98.40%	99.63%	99.90%	99.93%	6.30°(5.59°) ± 4.42°	99.53%
DT *(ours)*	**57.51%**	**89.40%**	**96.02%**	97.85%	98.87%	99.28%	99.40%	**5.76**°(**4.41**°) ± 6.98°	99.55%
DT_{r+s} *(ours)* †	56.30%	88.44%	95.23%	97.33%	98.68%	99.14%	99.27%	5.99°(4.47°) ± 7.53°	**99.64%**
DT_r *(ours)* ‡	51.48%	83.58%	92.87%	96.15%	98.10%	98.74%	98.94%	6.89°(4.86°) ± 8.84°	99.34%

⋆ The results of $PEDM$ are directly cited from [2].
† Refers to training the pose estimation network with both real and synthetic images.
‡ Refers to training the pose estimation network solely with real images.
* f subscript denotes "fake" real images.

and the decoder networks, to reduce the network parameter number. We also replaced the fully connected bottleneck layer with a convolutional one, which has been proven to stabilize the training of deep convolutional GANs [50]. Furthermore, a 1×1 convolutional layer was added before the final activation of the decoder to allow the generation of sharper synthetic output images. The discriminator D utilizes a simple CNN architecture consisting of four convolution-BatchNorm-LeakyReLU blocks [51]. The key feature here is that, apart from the binary output responsible for deciding between "real" and "fake" image samples, an extra quaternion regression layer [4] is added to enable 3D object pose quaternion estimation. Finally, the 3D object pose estimation network has the same architecture as in [1]–[4], where the number of input layer channels depends on the chosen operation mode (RGB or RGB-D).

In the conditional GAN training framework, the Adam optimizer [52] is used both for G and D with initial learning rate of 0.0001. Also, $\lambda_1 = 100$ and $\lambda_2 = 0.5$ were used in all the proposed RGB and RGB-D models. The 3D object pose estimation network hyperparameters are the same as the ones used in [4]. All experiments are conducted using Keras [53] with Tensorflow [54] backend, on an *Ubuntu* machine equipped with a *GeForce GTX 1080 Ti* graphics card.

B. Evaluation

In this subsection, the used datasets along with the adopted evaluation metric and all reported models are first described. Subsequently, all results for both the quantitative and the qualitative evaluation of the proposed framework are presented. Finally, the image synthesis performance of the proposed pose-oriented I2I network is discussed.

1) Experimental protocol: In all experiments reported in Tables II, III and V, the model of the proposed 3D object pose estimation framework as well as all competing models were trained using the Cropped Linemod dataset [1], [55]. The Cropped Linemod dataset consists of RGB-D sequences of 15 texture-less objects annotated with their 6D poses. In each image, the object of interest is centered and cropped to its 64×64 pixel ROI using the camera intrinsic parameters provided by the dataset. Following the same procedure as in [4], the real world RGB-D sequences of all objects in the Cropped Linemod dataset were roughly 50%-50% split over the training and test set, by ensuring a uniform viewpoint distribution over the viewing domain. Therefore, RGB-D sequences of the 15 objects were used both during training and testing, by also ensuring that neither of the networks (G, D, F, Q) has knowledge of the test RGB-D sequences during training. Both $\mathcal{S}_{i2i}$ and $\mathcal{S}_{db}$ sets are constructed using the real object images

from the training set and the 3D mesh models provided in the Cropped Linemod dataset. The protocol to collect $\mathcal{S}_{i2i}$ is the following. First, a random real object image $\mathbf{x}_i$ from the training set is selected. Then, using the corresponding ground truth 3D object pose $\mathbf{q}_i$ and the available 3D mesh models, a synthetic object image $\mathbf{y}_i$ is rendered under the same pose with plain background and is subsequently centered and cropped using its ground truth 64×64 pixel ROI, composing a $\mathcal{S}_{i2i}$ sample s_{i2i}. In total, $\mathcal{S}_{i2i}$ set consists of 8160 pairs of real and synthetic object images. The $\mathcal{S}_{db}$ set, similar to [1]–[4], consists solely of synthetic object images acquired by rendering the provided 3D object models by placing a virtual camera at 301 evenly distributed viewpoints on a half dome over the object. Each resulting image $\mathbf{y}_i$ is stored along with its object identity label c_i and its ground truth 3D object pose $\mathbf{q}_i$ to form a $\mathcal{S}_{db}$ sample $s_{db} = \{\mathbf{y}_i, c_i, \mathbf{q}_i\}$. In total, the $\mathcal{S}_{db}$ set consists of 15*301 = 4.515 s_{db} samples. Apart from the Cropped Linemod dataset, a second dataset was constructed, namely Cyclists, which consists of RGB images of an object with a more complex appearance (cyclist on a bicycle). This is also a challenging dataset as the same *cyclist* object appears in different scale, colors, and/or under occlusions. The Cyclists dataset is used only for a qualitative evaluation of the proposed framework, as all RGB images were manually annotated resulting in noisy 3D pose labels that do not allow fair quantitative evaluation. Note that the training and test sets, as well as $\mathcal{S}_{i2i}$ and $\mathcal{S}_{db}$ sets for the Cyclists dataset are acquired by following the same protocols as the ones used for the Cropped Linemod dataset.

All models are evaluated using the angular error in degrees between the ground truth object pose $\mathbf{q}$ and the predicted object pose $\hat{\mathbf{q}}$ [2], [4]:

$$err(\mathbf{q}, \hat{\mathbf{q}}) = 2 \arccos(|\mathbf{q}^T \hat{\mathbf{q}}|). \quad (9)$$

Given the angular error in degrees, the 3D object pose estimation accuracy is calculated only for test samples that are correctly classified to their object identity labels, as follows: 3D object pose estimation accuracy at threshold t is defined as the percentage of test samples, for which the angular error between the estimated and the ground truth 3D pose is below a threshold angle t, $err(\mathbf{q}, \hat{\mathbf{q}}) < t$ [2], [4].

The effectiveness of the proposed 3D object pose estimation framework is first evaluated by performing a comparison between the proposed method and the methods of [1]–[4], [20]. Note that, in contrast to our approach, [1]–[4] address the 3D object pose estimation problem by assuming a single domain for real and synthetic images and [20] by learning domain-invariant features. In addition, the proposed framework is compared against [17], [28], where domain adaptation is performed on pixel level, by generating "fake" real object images to train the 3D object pose estimation network. All trained models are evaluated in the Cropped Linemod test set, consisting of real RGB/RGB-D sequences that were not used during training as described above. We report all cases where the 3D object pose estimation network was trained a) solely on synthetic images (DT), b) on a mixture of real and synthetic images (DT_{r+s}) and c) solely on real images (DT_r). In addition, we implemented the $PGFL$ [3] and FM [20] methods, directly cited the results for RGB-D setting reported in [2] for $PEDM$ and used existing code for $3DPOD$ [1] and QL [4]. $PixelDA$ [17] denotes the model obtained by adapting the method of [17] to our network architectures, for a fair comparison. Moreover, we also modified [17] to train the 3D object pose estimation network using "fake" real and real images instead of "fake" real and synthetic images, denoted by $PixelDA_{f+r}$ [17]. This is to examine if labeled real data would lead to increased 3D object pose estimation performance. Finally, $PixelDA_{f+s+r}$ [17], [28] is obtained by blending the methods of [17], [28]. That is, the 3D object pose estimation network is trained using not only "fake" real and synthetic images, but also labeled real images, which corresponds to the semi-supervised setting in [28].

TABLE III
3D OBJECT POSE ESTIMATION ACCURACY FOR EACH OBJECT IN THE LINEMOD DATASET, FOR ANGLE THRESHOLD $t = 5°$ AND BOTH RGB AND RGB-D SETTING.

	RGB			RGB-D		
Objects	$PGFL$ [3]	QL [4]	DT	$PGFL$ [3]	QL [4]	DT
ape	33.53%	32.56%	**46.41%**	35.09%	34.16%	**55.49%**
benchv	34.57%	32.23%	**40.23%**	33.30%	35.14%	**46.35%**
bowl	72.74%	97.70%	**99.13%**	96.33%	97.26%	**99.71%**
cam	36.32%	33.43%	**47.57%**	39.42%	38.54%	**59.94%**
can	35.42%	32.40%	**40.26%**	36.24%	34.43%	**53.58%**
cat	32.35%	35.80%	**43.08%**	43.09%	39.81%	**54.52%**
cup	33.14%	**39.91%**	36.82%	36.09%	**39.86%**	35.82%
driller	29.87%	27.23%	**45.45%**	29.02%	24.69%	**52.11%**
duck	37.68%	31.06%	**50.07%**	32.50%	33.09%	**53.70%**
eggbox	37.13%	49.02%	**53.07%**	38.19%	47.08%	**63.56%**
glue	34.41%	48.66%	**58.54%**	42.15%	51.34%	**61.03%**
holep	29.74%	31.71%	**53.26%**	38.21%	40.54%	**59.29%**
iron	35.71%	34.58%	**53.97%**	40.28%	32.57%	**52.85%**
lamp	36.82%	34.00%	**47.01%**	36.05%	34.44%	**50.15%**
phone	37.82%	37.72%	**48.19%**	41.15%	33.67%	**61.75%**

Objects with **bold** text have shape symmetry.

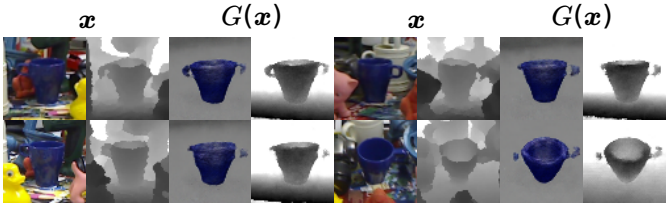


Fig. 4. Fail cases of the pose-oriented I2I network for the *cup* object.

2) Quantitative evaluation: The comparison of the 3D pose estimation accuracy for threshold angle values $t \in \{5°, 10°, 15°, 20°, 30°, 40°, 45°\}$ as well as the object classification accuracy between the proposed framework and all competing methods is presented in Table II. The object classification accuracy is reported for two reasons. First, since all competing methods are trained to simultaneously predict 3D object poses and object identities, object classification accuracy demonstrates the effect of the proposed framework in the

TABLE IV
INFERENCE TIME OF ALL COMPETING METHODS, CALCULATED USING A GEFORCE GTX 1080 TI GRAPHICS CARD.

	$3DPOD$ [1]	$PEDM$ [2]	$PGFL$ [3]	QL [4]	FM [20]	$PixelDA$ [17]
Inference time (ms)	5.25	5.25	5.25	5.25	86.48	5.25
	$PixelDA_{f+r}$ [17]	$PixelDA_{f+s+r}$ [17], [28]		DT *(ours)*	DT_{r+s} *(ours)*	DT_r *(ours)*
Inference time (ms)	5.25	5.25		11.25	11.25	11.25

classification task. Second, similar to [24], object classification accuracy serves as a way to evaluate the proposed I2I network. This is because the objective of the proposed I2I network is to generate high-quality, noise-free synthetic images where the depicted objects should be perfectly recognizable and therefore, object classification should be easier to perform. The results reported in Table II show that the proposed framework DT significantly outperforms all methods for the high accuracy threshold angle values $t = 5^\circ, 10^\circ$, in both RGB and RGB-D settings. When the proposed method (DT) is compared against methods that assume a single domain for real and synthetic images ($3DPOD$, $PEDM$, $PGFL$, QL), the 3D object pose estimation accuracy is increased up to 17%. This can be explained by the fact that the final 3D object poses are easier to discriminate in the translated synthetic image domain, which is clear from background clutter and other possible distractions. In addition, the comparison between DT and FM, where 3D object pose estimation accuracy is increased over 18%, shows that solving the 3D object pose estimation problem in the synthetic image domain by translating real images to synthetic ones during inference is a superior approach compared to learning domain-invariant features. Moreover, the 3D object pose estimation accuracy of DT is also significantly increased for the threshold angle values $t = 5^\circ, 10^\circ$ when compared to the ones of $PixelDA$, $PixelDA_{f+r}$ and $PixelDA_{f+s+r}$, proving that the proposed framework is far more effective for 3D object pose estimation. DT outperforms the best performing model of [17] ($PixelDA_{f+s+r}$) up to 14% even though $PixelDA_{f+s+r}$ employed "fake" real and real objects images in addition to synthetic images during 3D object pose estimation training. On the other hand, the proposed framework displayed similar performance with all other methods in the medium and low 3D object pose accuracy thresholds $t \in \{15^\circ, 20^\circ, 30^\circ, 40^\circ, 45^\circ\}$. The latter can be explained by errors introduced in the I2I step, rather than inaccuracies of the actual 3D object pose estimation step. The results reported in Table II also show that the proposed method demonstrates lower mean angular error values for both RGB and RGB-D settings as well as higher object classification accuracy, which proves that the proposed framework also benefits object classification and that the I2I network was able to successfully translate real images to synthetic ones according to its objective.

Furthermore, the comparison between DT_r and QL, which use the same 3D object pose estimation network, shows that performing domain translation on the test real object images as a preceding step significantly increases the 3D object pose estimation accuracy for the high accuracy threshold angle value $t = 5^\circ$, even in the case where the 3D object pose estimation network is trained only on noisy real images. This also shows that the 3D object pose estimation network is able to more accurately perceive 3D object pose differences on the translated synthetic images. However, the 3D object pose estimation accuracy of DT_r is inferior when compared to the cases where synthetic images are used exclusively (DT) or in combination with real images (DT_{r+s}) to train the 3D object pose estimation network. These results further strengthen our statement that clean synthetic images are more suitable for learning pose-related image features than noisy real images. Finally, the effect of training the 3D object pose estimation network in the proposed framework using only synthetic images or a mixture of real and synthetic images can be seen by comparing the pose estimation performance between DT and DT_{r+s} in Table II. Since the 3D object pose estimation accuracy is similar at all threshold angle values both in RGB and RGB-D, with the accuracy differences being below 1.2%, it is proven that including real images in the training set does not provide any extra beneficial information to the 3D object pose estimation network.

To further evaluate the proposed 3D object pose estimation framework, the 3D object pose estimation accuracy at the high accuracy threshold angle value $t = 5^\circ$ for each object in the Cropped Linemod dataset is presented in Table III. Note that, similarly to [1], [4], we also treat the objects *bowl* and *cup* as rotationally invariant, and the *eggbox* and *glue* objects as 180° symmetric around the z-axis. As can be seen in the reported results, the proposed framework DT improves the 3D object pose estimation accuracy by a large margin for all objects, with an improvement over 10% in most objects and both in RGB and RGB-D settings. Specifically in the cases of the *ape* and *cam* objects in the RGB-D setting and the *holepuncher* object in the RGB setting, the improvement exceeds 20%. The only exception is the *cup* object, where the 3D object pose estimation accuracy is below $PGFL$ and QL, due to the accumulated error from improperly executed image-to-image translation. Such a case is presented in Fig. 4, where the *cup* image-to-image translation fails, mostly because the handle appears in more than one side of the cup.

Finally, the execution time of all competing methods are presented in Table IV. The execution time of $3DPOD$, $PEDM$, $PGFL$, QL, $PixelDA$ and its variants ($PixelDA_{f+r}$, $PixelDA_{f+s+r}$) involves the forward pass through the feature extraction network F and the NN search between the extracted image feature vector $\mathbf{f}$ and the precalculated database image features $\mathbf{f}_i$. In all these cases, the same feature extraction network architecture is used (F), thus the execution time is the same as well, calculated at 5.25 ms. FM uses a deeper architecture (a variant of VGG-16 [56])

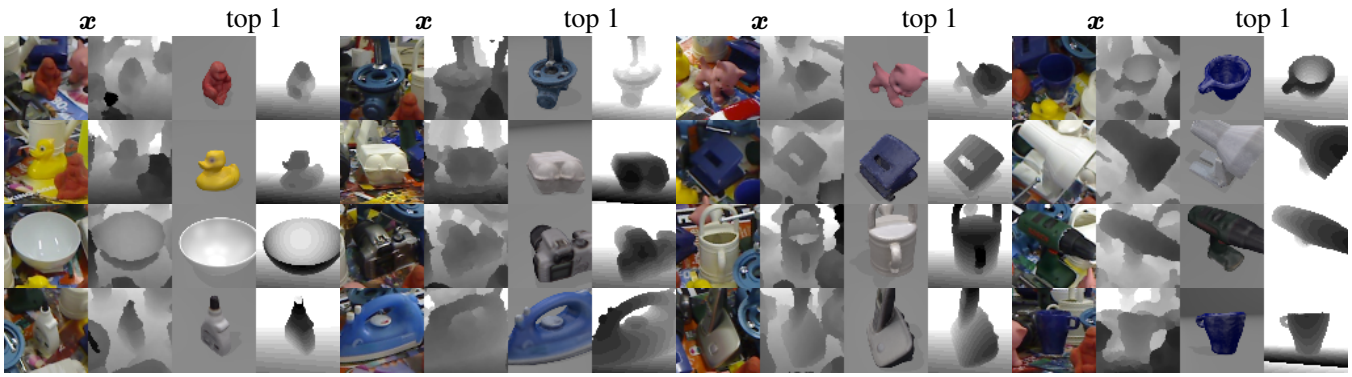


Fig. 5. Examples of test RGB-D images and the corresponding closest database samples retrieved by the NN search algorithm under the proposed 3D object pose estimation framework.

for the feature extraction network, requiring an inference time of 86.48 ms. The proposed method includes an extra image-to-image translation inference step before the feature extraction network F to translate real object images to synthetic ones, as detailed in Subsection III-B. The execution time of this extra step was calculated at 6 ms, meaning that the total inference time of DT, DT_{r+s} and DT_r is 11.25 ms. As can be observed, this increase of complexity doubles the inference cost (compared to using only F), nevertheless, the proposed method remains very fast and executes at over-real-time speed, especially when compared with the inference cost of a deeper feature extraction network architecture like the one used in FM. Most importantly, this additional inference step of the proposed method significantly increases the 3D object pose estimation performance as presented in Tables II and III. Note that for all time calculations a *GeForce GTX 1080 Ti* graphics card was used.

3) Qualitative evaluation: Apart from the quantitative evaluation, we also perform a qualitative evaluation of the proposed 3D object pose estimation framework. This is to offer a more intuitive demonstration of the proposed 3D object pose estimation framework performance. Real query images of all objects in the Cropped Linemod dataset along with the corresponding top 1 retrieved database images using the proposed 3D object pose estimation framework are presented in Fig. 5. In all cases, the 3D pose difference between the query images and the database images retrieved by the proposed framework are imperceptible. The last example shown in Fig. 5 is of particular interest as the cup handle in the retrieved database image appears in the opposite side of the cup with respect to the query image. However, this is expected as we treated the *cup* object as rotationally invariant, meaning that different 3D poses correspond to different elevation values only. This proves that the proposed framework have successfully learned to handle symmetric objects. Moreover, the performance of the proposed framework is evaluated on the Cyclists dataset. Test real *cyclist* images along with the corresponding translated synthetic images obtained by the proposed I2I network and the top 5 retrieved database images are depicted in Fig. 6. The 3D poses of the retrieved database samples are almost identical to the ones of the test images, even in the cases where the *cyclist* object is not clearly seen. In fact, the 3D object pose estimation framework is still able to predict accurate 3D object poses even from non-perfectly translated synthetic images. Such cases can be seen in Fig. 6, in rows 6 and 7, where although the I2I network has introduced some noise in the translated synthetic images, all top 5 retrieved database images depict the *cyclist* object in 3D poses that are very similar with the ones in the query images.

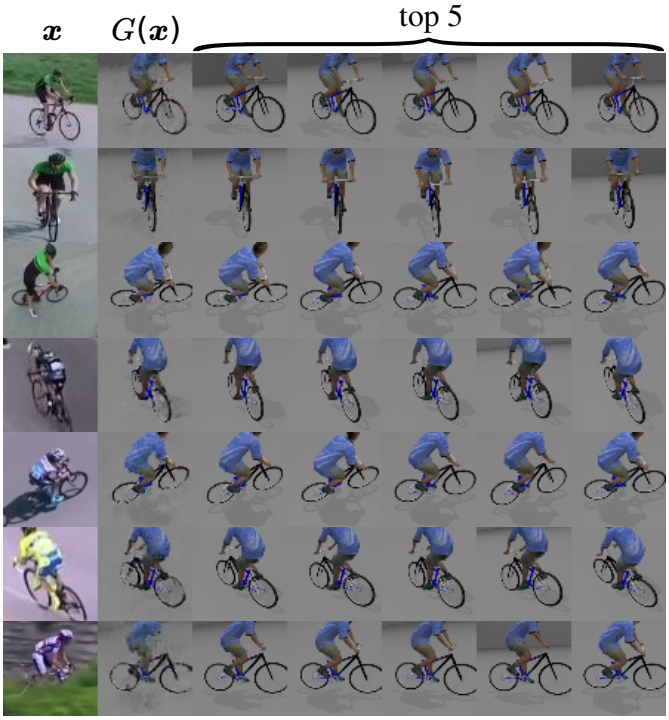


Fig. 6. Image-to-image translation results and retrieved top 5 nearest neighbors for 7 query *cyclist* images. The first and second columns show the query images and the image-to-image translation results, respectively. The rest columns depict the retrieved closest nearest neighbors from left to right. Although the image-to-image translation visual quality is not optimal in every case, this does not hinder the performance of the 3D object pose estimation network, as the necessary pose-related characteristics have been accurately preserved.

4) Evaluation of image-to-image translation: The image synthesis performance of the proposed pose-oriented I2I network for both the Cropped Linemod and the Cyclists datasets is also evaluated by measuring the reconstruction error be-

TABLE V
3D OBJECT POSE ESTIMATION AND OBJECT CLASSIFICATION ACCURACY FOR DIFFERENT VARIATIONS OF THE PROPOSED FRAMEWORK. DT_{PGFL} DENOTES THAT THE 3D OBJECT POSE ESTIMATION NETWORK IN THE PROPOSED FRAMEWORK IS REPLACED WITH THE ONE INTRODUCED IN [3] AND $DT_{\lambda_p=0}$ THAT THE I2I NETWORK IN THE PROPOSED FRAMEWORK IS TRAINED WITHOUT (6) IN (7). FINALLY, DT_{AE} DENOTES THAT THE DISCRIMINATOR NETWORK D IS COMPLETELY REMOVED DURING I2I NETWORK TRAINING AND DT_{max} THAT THE 3D OBJECT POSE ESTIMATION NETWORK IS EVALUATED ON THE GROUND TRUTH SYNTHETIC OBJECT IMAGES.

	Angular threshold t				**Object classification**
	5°	**10°**	**20°**	**40°**	
			RGB		
DT	**51.08%**	**83.97%**	**97.29%**	**99.43%**	**99.01%**
DT_{PGFL}	41.47%	71.31%	87.95%	94.92%	98.49%
$DT_{\lambda_p=0}$	49.12%	81.96%	96.61%	99.25%	98.68%
DT_{AE}	49.42%	82.05%	96.55%	99.24%	98.82%
DT_{max}	71.37%	96.50%	99.96%	100%	100%
			RGB-D		
DT	**57.51%**	**89.40%**	**97.85%**	**99.28%**	99.55%
DT_{PGFL}	43.83%	73.14%	88.48%	95.50%	98.75%
$DT_{\lambda_p=0}$	55.56%	88.52%	98.04%	99.18%	99.29%
DT_{AE}	55.42%	88.22%	97.81%	99.21%	**99.63%**
DT_{max}	71.59%	96.91%	99.97%	100%	100%

tween the target and the generated images. Similar to [18], the reconstruction error is defined as $\|\mathbf{y} - G(\mathbf{x})\|_1$, where $\mathbf{y}$, $G(\mathbf{x})$ is the synthetic target image and the translated synthetic image produced by the proposed I2I network, respectively. The reconstruction error is calculated only for the test sequences of both datasets, being 0.2028 for the Cropped Linemod dataset and 0.1994 for the Cyclists dataset. Finally, Fig. 3 and Fig. 6 also offer the opportunity to visually evaluate the performance of the proposed I2I network for both datasets. As can be seen in both figures, the proposed I2I network is able to generate synthetic images of sufficient quality in all cases. However, as explained in Subsection III-B, the proposed framework does not insist on achieving optimal visual quality for the image-to-image translation, as long as the specific pose characteristics are preserved in the translated images. Such cases are presented in Fig. 6, in rows 6 and 7.

C. Ablation study

In order to investigate the contribution of each component of the proposed 3D object pose estimation framework, an ablation study has been performed. Note that, in all cases in the ablation study, the 3D object pose estimation network used in the proposed framework is trained solely on synthetic images, similar to DT. First, we examine the contribution of the specific 3D object pose estimation network [4], that was adopted in the proposed framework, to the 3D object pose estimation performance. To this end, we replaced our adopted 3D object pose estimation network with the one proposed in [3], which equals to training the 3D object pose estimation network using the loss function proposed in [3] (as the 3D object pose estimation network in [3] has the same architecture as the one we use in our work) that can also be seen in Table I. We will refer to this new model as DT_{PGFL}, hereafter. By examining the results reported in Table V, it is shown that the 3D object pose estimation accuracy of DT_{PGFL} is inferior to the performance of the proposed DT for all angle threshold values $t \in \{5°, 10°, 15°, 20°, 30°, 40°, 45°\}$. The most obvious explanation is that DT_{PGFL} overfits to clean synthetic images during training and, thus, cannot generalize to the translated synthetic images, due to the added background clutter. In contrast, the adopted 3D object pose estimation method is more robust to noisy inputs, making it a more suitable choice for the proposed framework. Nevertheless, by comparing the pose estimation performance between $PGFL$ and DT_{PGFL} reported in Tables II and V, respectively, it can been seen that when the same pose estimation network ($PGFL$) is utilized in the proposed framework, the pose estimation performance at the high accuracy threshold $t = 5°$ increases.

The effect of training the discriminator network D not only to validate the outputs of the I2I network but to also perform 3D object pose regression can be seen by omitting the quaternion regression term (6) from (7) during the I2I network training. Note that this equals to setting $\lambda_p = 0$ in (7) and therefore, the derived trained model in this case is denoted by $DT_{\lambda_p=0}$. As presented in Table V, by removing the $\mathcal{L}_p(D)$ term from the objective function the pose estimation accuracy reduces, as the absence of the $\mathcal{L}_p(D)$ term in (7) results in a weaker discriminator, which in turn causes the I2I network to be insufficiently trained and to output translated images of lower quality. Finally, in order to investigate the effect of using the discriminator D to validate the outputs of the I2I network in the conditional GAN training framework, we completely removed the discriminator network D during training. That is, the I2I network is trained to simply reconstruct synthetic object images according to (5). Essentially, G in this case is trained similar to an autoencoder [38] and so, the corresponding trained model is denoted by DT_{AE}. The reported 3D object pose estimation accuracy of DT_{AE} shows that removing the discriminator and using only (5) to train the I2I network also hurts the final pose estimation performance when compared to DT. However, this result is expected as GANs have been proven [24], [45] to further increase the quality of the generated images relative to simply using the mean squared error (5). Interestingly, DT_{AE} achieves similar performance to $DT_{\lambda_p=0}$, proving that if the discriminator is not effectively trained ($DT_{\lambda_p=0}$), the conditional GAN training framework does not improve the performance of the proposed framework.

The contribution of the I2I network and the 3D object pose estimation network to the final 3D object pose estimation performance can be examined by evaluating the trained 3D object pose estimation network directly on the ground truth synthetic images $\mathbf{y}$, instead of the translated synthetic images $G(\mathbf{x})$ that are obtained from the I2I network (normal mode in the proposed framework). Note that this setting coincides

with the top performance that the proposed framework can achieve with the given 3D object pose estimation network, as it simulates a perfect I2I network and will be referred as DT_{max}. The contribution of the 3D object pose estimation network to the final performance can be seen by directly examining the results of DT_{max} in Table V, as this is the only network that is applied on the ground truth synthetic images. DT_{max} achieves 71.37% and 71.59% 3D object pose estimation accuracy at the high accuracy threshold $t = 5^\circ$ in the RGB and RGB-D settings, respectively, while the accuracy is over 96% for the threshold values $t = 10^\circ, 20^\circ$ and 100% at the threshold $t = 40^\circ$. Although the 3D object pose estimation network is not able to perfectly estimate 3D object poses for all threshold values, the 3D object pose estimation performance of DT_{max} at the high accuracy threshold $t = 5^\circ$ is dramatically increased or even doubled when compared to the ones of all competing methods (Table II), proving again that the 3D object pose estimation problem is easier to solve in the synthetic image domain.

Moreover, as the only difference between DT_{max} and DT is that the 3D object pose estimation network acts on the ground truth synthetic object images (DT_{max}) instead of the translated synthetic object images (DT), the comparison between DT_{max} and DT allows us to evaluate the contribution of the I2I network to the final 3D object pose estimation performance. This comparison implies that any difference between the performance of DT_{max} and DT is caused by the errors introduced by the I2I network in the translated synthetic object images, which in turn undermine the final 3D object pose estimation performance. By examining the results presented in Table V, the introduced error of the proposed I2I network in the final performance can be measured by calculating the difference between the performance of DT_{max} and DT, which is 20% and 14% at the high accuracy threshold $t = 5^\circ$ in the RGB and RGB-D settings, respectively. These results also show that the proposed framework has some limitations caused either by the I2I network or by the 3D object pose estimation network architectures. Limitations of the latter are shown by the fact that the 3D object pose estimation network cannot achieve 100% accuracy for all threshold values even on the ground truth synthetic object images. Limitations are also caused by the image-to-image translation step, i.e., when the image-to-image translation fails, the final 3D object pose estimation accuracy degrades. Such is the case of the *cup* object, where some details e.g., the position of the handle, are not estimated correctly by the I2I translation network. Since this object is considered as rotationally invariant in the dataset, the position of the handle appears in either the left or the right side of the cup, without affecting its ground truth 3D pose. As a result, the image-to-image translation network has learnt that the position of the handle is not important, thus in some cases it outputs images where the handle appears in more than one side of the cup, generating in essence an object that does not look like a cup (Fig. 4). Nevertheless, both these limitations could possibly be addressed by replacing the 3D object pose estimation and I2I networks with more powerful ones, as the modular design of the proposed framework allows any network to be used in the image-to-image translation and the pose estimation steps. However, in this case the inference time would probably increase considerably. Another possible solution to the limitations of the I2I network may be to explicitly or implicitly train it to specifically handle such difficult (*cup* object) cases.

V. Conclusion

In this work, a novel 3D object pose estimation framework which consists of a pose-oriented image-to-image translation and a 3D object pose estimation network was presented. By first translating noisy real object images to clean synthetic ones and then applying the pose estimation network on the translated images, the proposed framework demonstrated significantly increased 3D object pose estimation accuracy in the high accuracy area ($t = 5^\circ, 10^\circ$) when compared to state-of-the-art object pose estimation methods. As the final 3D object poses in the proposed framework are estimated using the translated synthetic images that are obtained by the proposed image-to-image translation network, the 3D object pose estimation network can be trained solely on synthetic images. Therefore, the proposed framework offers a reliable solution to the problem of lack of accurately annotated data. Moreover, the proposed pose-oriented objective function used in the conditional GAN training framework ensured that the specific pose-related object image characteristics are preserved in translated images, allowing the 3D object pose estimation network to more accurately predict 3D poses.

Future work includes extending this method by training the image-to-image translation network to generate clean synthetic images from real images where objects are heavily occluded or where objects appear in different background scenes. Moreover, the 3D object pose estimation framework can be extended to internally handle object symmetries.

References

[1] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 3109–3118.

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

[3] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose guided rgbd feature learning for 3d object pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3856–3864.

[4] C. Papaioannidis and I. Pitas, "3d object pose estimation using multi-objective quaternion learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[5] M. Bui, S. Zakharov, S. Albarqouni, S. Ilic, and N. Navab, "When regression meets manifold learning for object recognition and pose estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[6] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3828–3836.

[7] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold *et al.*, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 3364–3372.

[8] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7668–7677.

[9] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1941–1950.

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

[11] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1329–1335.

[14] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2686–2694.

[15] S. Mahendran, H. Ali, and R. Vidal, "3d pose regression using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2174–2182.

[16] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 205–220.

[17] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 3722–3731.

[18] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *European Conference on Computer Vision*, 2018, pp. 35–51.

[19] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in neural information processing systems*, 2016, pp. 343–351.

[20] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3d pose inference from synthetic images," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 4663–4672.

[21] ——, "Domain transfer for 3d pose estimation from color images without manual annotations," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 69–84.

[22] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[23] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *European Conference on Computer Vision*, 2018, pp. 699–715.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[27] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International journal of computer vision*, pp. 1–16, 2020.

[28] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 4500–4509.

[29] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 1510–1519.

[30] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2938–2946.

[31] T. Chen and S. Lu, "Robust vehicle detection and viewpoint estimation with soft discriminative mixture model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 394–403, 2015.

[32] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[33] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[34] T.-T. Do, M. Cai, T. Pham, and I. Reid, "Deep-6dpose: Recovering 6d object pose from a single rgb image," *arXiv preprint arXiv:1802.10367*, 2018.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

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

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[39] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[41] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.

[42] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[43] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[44] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[45] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[46] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with pixelcnn decoders," in *Advances in neural information processing systems*, 2016, pp. 4790–4798.

[47] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[49] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2642–2651.

[50] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[53] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[54] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar,

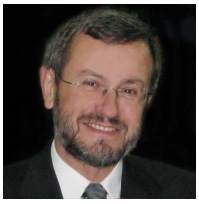
P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[55] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 548–562.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

Christos Papaioannidis received his Diploma in Electrical & Computer Engineering in 2015 from the Aristotle University of Thessaloniki. He is currently pursuing his Ph.D. studies in the Artificial Intelligence & Information Analysis Laboratory in the Department of Informatics at the Aristotle University of Thessaloniki. His research interests include deep learning and image analysis.

Vasileios Mygdalis received the B.Sc. degree in Biomedical Informatics in 2010 and the M.Sc. degree in Medical Informatics in 2014, from the University of Central Greece and Aristotle University of Thessaloniki, Greece, respectively. He is currently a researcher and teaching assistant and he is studying towards a PhD at the Department of Informatics at the University of Thessaloniki. He has co-authored more than 25 papers in academic journals and international conferences. His research interests include machine learning, image/video processing, computer vision and pattern recognition.



Prof. Ioannis Pitas (IEEE fellow, IEEE Distinguished Lecturer, EURASIP fellow) received the Diploma and PhD degree in Electrical Engineering, both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics of the same University. He served as a Visiting Professor at several Universities. His current interests are in the areas of image/video processing, intelligent digital media, machine learning, human centered interfaces, affective computing, computer vision, 3D imaging and biomedical imaging. He has published over 861 papers, contributed in 44 books in his areas of interest and edited or (co-)authored another 11 books. He has also been member of the program committee of many scientific conferences and workshops. In the past he served as Associate Editor or co-Editor of eight international journals and General or Technical Chair of four international conferences. He participated in 69 R&D projects, primarily funded by the European Union and is/was principal investigator/researcher in 41 such projects. He has 27310+ citations (Source Publish and Perish), 8216+ (Scopus) to his work and h-index 80+ (Source Publish and Perish), 44+ (Scopus).