HOW TO TRACK YOUR DRAGON: A MULTI-ATTENTIONAL FRAMEWORK FOR REAL-TIME RGB-D SINGLE OBJECT POSE TRACKING

Isidoros Marougkas[1], *Petros Koutras*[1], *Nikos Kardaris*[1],
Georgios Retsinas[1], Georgia Chalvatzaki[3], Petros Maragos[1]

[1]School of E.C.E., National Technical University of Athens, 15773, Athens, Greece
[2]School of E.C.E., National Technical University of Athens, 15773, Athens, Greece
[3]School of E.C.E., National Technical University of Athens, 15773, Athens, Greece
Email: ismarougkas@gmail.com, {pkoutras, gretis, nkardaris, gchal, maragos}@cs.ntua.gr

ABSTRACT

In this work, we propose an extension of previous convolutional architectures, with the goal of real-time RGBD - based Object Pose Tracking. We explicitly handle a variety of challenges, arising by the problem's formulation, which were previously left for the network to figure out by itself. Majorly, we tackle background clutter and occlusions by employing multiple parallel Soft Spatial Attention modules, we sophisticate the data augmentation procedure and we take into consideration the geometrical properties of both the object's $3D$ model and the Pose Space when training the network. As a result, we improve the State-of-the-Art (SoA) tracking performance by an average score of 76%, when testing on the hardest available dataset up to date.

***Index Terms*—** Pose, Tracking, Attention, Geodesic, Multi-Task

1. INTRODUCTION

The attempt to estimate the pose of an object from an RGBD image pair faces the innate challenges of background clutter, occlusions (both static, from other objects present in the scene, and dynamic, due to possible interactions with a human user), illumination variation, sensor noise, image blurring (owing to fast movement) and appearance changes, as the object viewpoint alters. Moreover, we should account for the pose ambiguity, that emanates for the object's own geometry, in possible symmetries, the challenges of proper parameter representation of rotations and the inevitable difficulties that an effort to forge a model faces, when extracting information about the 3D scene geometry from 2D-projected images.

In summary, the main contributions of our work are:

- An explicit background clutter and occlusion handling mechanism that leverages Soft Spatial Attentions, and provides an intuitive understanding of the tracker's region of interest at each frame, while boosting its performance.
- The use of a Multi-Task Pose Tracking loss function, that respects the geometry of both the object's 3D model and the Pose Space and boosts the tracking performance, by optimizing auxiliary tasks along with the principal one.
- Achieving new, SoA Real-time performance in the hardest scenario of the benchmark dataset of [2] lowering both translation and rotation errors by an average margin of 76%.

2. RELATED WORK

2.1. Tracking-by-Detection

In PoseCNN [**?**], the authors constructed a CNN that estimates binary object masks and then predicts the object class, translation and rotation seperately. In SSD6D [**?**], they extended the SSD framework [**?**] for 2D Object Detection in order to perform dicrete viewpoint classification for some known object cases. Later, they refined their initial estimations with ICP[**?**] iterations. DPOD [**?**] inputs RGB images to a ResNet[**?**] encoder to segment a pixel-wise prediction of the object identification in a mask-level. Following this, UV texture maps are estimated to extract dense Object Correspondences between 2D images and 3D object models minimizing Cross Entropy Losses. Finally, those correspondences are used for Pose Estimation via P'n'P[**?**]. This estimation is, then, inserted as a prior to a refinement CNN that outputs the final pose prediction. Lastly, iPose [**?**] is one of the at-

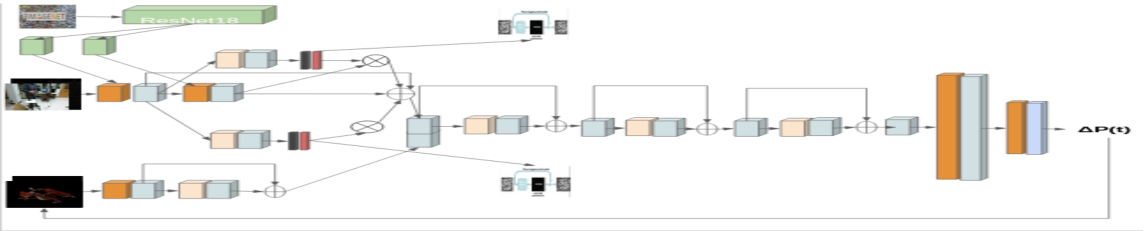

Fig. 1: Overview of the proposed approach. The first two layers of the Observed convolutional stream are initialized with the weights of Resnet18 [**?**] pretrained on Imagenet to narrow down the real-synthetic domain adaptation gap, as proposed in [**?**]. To the output of the second Observed layer we apply the learnable Spatial Attention Maps for Foreground Extraction and Occlusion Handling (that we describe in detail in Sec.3.3.1) and we add their corresponding output feature maps with the one of the second layer, along with a Residual connection [**?**] from the first layer.

tempts closest to ours. The authors segment binary masks with a pretrained MaskRCNN [**?**] to extract background clutter and occluders and, with an Encoder-Decoder they map 2D pixels to dense 3D Object Coordinates, which, in turn, are used as input to a P'n'P [**?**] geometric optimization. Our Attention modules have the same effect, but are computationally cheaper.

2.2. Temporal Tracking

In the baseline our work extends, Garon et.al [1],[2] formulated the tracking problem exclusively as a learning one by generating synthetic RGBD frame pairs from independent viewpoints and regress the pose using a CNN. In DeepIM [**?**], they initialized a similar CNN with the weights of a FlowNet2 [**?**] and fused its two streams by subtraction. In DeepTAM [**?**], they used an Optical flow-based term for regularization during training and encourage the production of multiple heterogenous pose hypotheses that got bootstrapped in the final layer.

3. METHODOLOGY

3.1. Data Augmentation

We modify the augmentation procedure of [1],[2] as follows: Firstly, we blend the object image with a background image, sampled from a subset of the SUN3D dataset [**?**]. We also mimic [1],[2] in rendering a 3D hand model-occluder on the object frame with probability 60%. A twist we add is preparing our network for cases with 100% occlusion, by completely covering the object by the occluder for 15% of the occluded subset. Note that both the foreground and unoccluded object binary masks ($M_{Foregr/Occl}$) are kept during both of these augmentation procedures. Hence, we can use them as ground truth segmentation signals for clutter extraction and occlusion handling in our auxiliary losses, to supervise the corresponding Spatial Attention maps. We also add to the "Observed" frame pair $I(t)$: (i) Gaussian RGB noise, (ii) HSV noise, (iii) blurring (to simulate rapid object movement), (iv) depth downsampling and (v) probabilistic dropout of one of the modalities, all with the same parameters as in [2] . With a probability of 50%, we change the image contrast, using parameters $\alpha \sim U(0,3)$, $\beta \sim U(-50,50)$ and gamma correction $\gamma \sim U(0,2)$ with probability 50%, to help generalize the performance to illumination variation conditions between render and sensor generated images. Also, instead of modelling the noise added to the "Observed" Depth modality with an ad-hoc Gaussian distribution as in [2], we consider the specific properties of Kinect noise [**?**] and model it with 3D Gaussian noise that varies with depth and angle. The rest of the preprocessing follows [1] .

3.2. Object Pose Mathematical Formulation

Usually, the object pose $\mathbb{P}$ is described as a rigid 3D Transformation w.r.t. a fixed coordinate frame, namely an element of the Special Euclidean Lie group in 3D: $SE(3)$. It can be disentangled into two components; a rotation matrix R, which is an element of the Lie Group $SO(3)$ and a translation vector $\mathbf{t} \in$R^3. However, Bregier et al. [**?**] proposed a broader definition, where the object pose is considered a family of such transformations, due to the ambiguity, inserted by a possible rotational symmetry, notated as G $\in$ SO(3). This leads to relaxing the Pose Space $\mathscr{C}$ definition:

$$\mathscr{C} = \Bigg\{ \mathbb{P} \quad | \quad \mathbb{P} = \left[\begin{array}{c|c} R \cdot G & \mathbf{t} \\ \hline \mathbf{0}^T & 1 \end{array} \right], \\ \mathbf{t} \in \mathbb{R}^3, R \in SO(3), G \in SO(3) \Bigg\}. \tag{1}$$

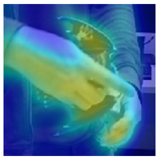

Fig. 2: A Spatial Attention weight map dedicated to foreground extraction. It focuses on the moving parts of the scene (i.e. the hand and the object) and gets blurier as the occlusion percentage increases.

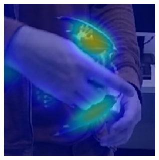

Fig. 3: A Spatial Attention weight map dedicated to occlusion handling. As the occlusion percentage increases, it gets sharper and shifts focus from the object center to its body parts.

As stated in [**?**], the description of the pose of an object with spherical symmetry requires just 3 numbers: $(t_{x,y,z})$, as G can be any instance of $SO(3)$ with the imprinted shape of the object remaining the same. It is obvious that, for asymmetric objects, G=$\mathbb{I}_3$ and the pose representation is unique.

3.3. Architecture Description

The full architecture of our network is shown in Fig.1.

3.3.1. Background and Occlusion Handling

After our first "Observed" Fire layer[], our model generates an Attention Weight map $Att_{Unoccl/Foregr}$, by using a convolutional Layer dedicated to occlusion handling and foreground extraction, respectively, followed by a 1×1 convolution that squeezes the feature map channels to a weight map (normalized by Softmax). Our goal is to distil the soft foreground and occlusion segmentation masks from the hard binary ground-truth ones, in order to have their estimations available during the tracker's inference. To this end, we add the two corresponding Binary Cross Entropy losses to our overall loss function. One may wonder why the occlusion map is not enough by itself to highlight the features of interest, as its goal is a subset of foreground extraction. We experimentally found that assigning a clear target to each of the two modules is more beneficial to the main one, than hoping that a single Attention layer will be able to handle both challenges at once.

3.3.2. Overall Loss and Rotation Representation

From a mathematical standpoint, immediate regression of pose parameters [2] with an Euclidean loss is suboptimal: the translation component is, indeed, an element of the Euclidean space but the rotation component lies on a non-linear manifold of $SO(3)$. It is straightforward to model the rotation loss using a Geodesic rotation metric of $SO(3)$ between $\hat{R}, R$ (see eq.2). In order to minimize the rotation errors due to ambiguities caused by the parameterization choice, we employ the 6D continuous rotation representation that was introduced in [**?**]. Furthermore, as it has already been discussed in [**?**], each 3D rotation angle has a different visual imprint regarding each rotation axis. So, we multiply both rotation matrices with a diagonal Inertia Tensor Λ calculated on the object model's weighted surface and w.r.t. its center mass, in order to assign a different weight to each rotational component. Since we want that matrix product to still lie in $SO(3)$, we perform a Gramm-Schmidt orthonormalization of Λ before multiplying it with each rotation matrix. Finally, we choose to weigh the translation and rotation losses using a pair of learnable weights $\mathbf{v} \in \mathbb{R}^2$ that are co-optimized via Adam, as proposed by [**?**]. Overall, our tracking loss function is formulated as:

$$\begin{aligned} L_{Track}(\hat{\Delta P}, \Delta P) = exp(-v_1) \cdot MSE[(\hat{\Delta \mathbf{t}}, \Delta \mathbf{t})] + v_1 \\ + \exp(-v_2) \cdot arcos\left\{ \frac{tr[(\hat{R}\Lambda_{(G.S.)})^T R\Lambda_{(G.S.)}] - 1}{2} \right\} + v_2 \end{aligned} \tag{2}$$

Using a similar external Multi-Task learnable weighting scheme ($\mathbf{s} \in \mathbb{R}^3$) as in Sect.3.3.2, we combine our primary learning task, pose racking, with the two auxiliary ones: clutter and occlusion handling:

$$\begin{aligned} Loss = exp(-s_1) \cdot L_{Track} + exp(-s_2) \cdot L_{Unoccl} + \\ exp(-s_3) \cdot L_{Foregr} + s_1 + s_2 + s_3 \end{aligned} \tag{3}$$

3.3.3. Symmetric Object Handling

In the special case of symmetric objects, we disentangle the ambiguities inserted due to this property from the core of rotation estimation. So, we regress a separate Euler angle triplet of symmetry-based parameters $\mathbf{g} \in \mathbb{R}^3$ that is converted to a rotation matrix G, which gets right-multiplied with R before being weighted by the parameters of $\Lambda_{(G.S.)}$. We used a cylindrical cookiejar model for the symmetric object case, the shape of which has only one axis of symmetry. So, we regress a single symmetry parameter, that of the object-centric z-axis. Before the conversion, that parameter is passed through a Tanh activation function and multiplied by π to constrain its values.

4. IMPLEMENTATION DETAILS

We use ELU activation functions, the Adam optimizer with corrected weight decay [**?**] by a factor $1e^{-5}$, a learning rate of $1e^{-3}$ and a scheduler with warm restarts [**?**] every 10 epochs. All network weights (except those transferred from ResNet18[**?**]) are initialized via a uniform Kaiming He [**?**] scheme. Since the Geodesic distance suffers from multiple local minima, following [**?**], we first warm-up the weights by aiming to minimize the LogCosh loss function for 25 epochs. Then, we train it until convergence, minimizing (eq.3).

5. EVALUATION AND RESULTS

5.1. Dataset and Metrics

We test our approach on the "hard interaction scenario" of [2]. It is the most difficult one contained in the dataset, as it is comprised by free 3D object motion, along with arbitrary occlusions by the user's hand. We assume that if our tracker performs best when we test on this, it will have the same behaviour in every other scenario. The metrics we use are the ones defined in [2] and we, also, initialize our tracker every 15 frames.

5.2. Experimental Results

5.2.1. The Dragon model: an asymmetric object

Our approach reduces both mean errors by about 76%, while it also significantly increases the tracker's robustness, as measured by the inverse of standard deviation of the errors (by $\%$ and $\%$, respectively). In particular, the approach of [2] achieves a translation error of 48.58 ± 38.23 mm and a rotation error of $35.43^0 \pm 34.74^o$, while ours improves them to 11.63 ± 8.79 mm/$8.31^o \pm 6.76^o$. When the object is unoccluded, the tracker focuses mostly on its 3D center, implicitly realizing in this way, that this is the main 3D point of tracking interest. When the user's hand occludes parts of the dragon, the Attention shifts to its body parts of interest that stand out of the grip, like its neck, wings or tail. We also observe a visual tradeoff between Attentions (see fig.**??**): while the object is unoccluded (either in steady state or when moving), the module responsible for foreground extraction is highlighted more intensely than the occlusion one. As the object gets more and more covered by the user's hand, the focus gradually shifts to the module responsible for occlusion handling. Note that this is not an ability we explicitly train our network to obtain, but rather a side effect of our approach, which fits our intuitive understanding of cognitive visual tracking. The effectiveness of our method is demonstrated by the fact that, while [1] states that their CNN keeps track only of the object's 3D position under heavy occlusions, our improved one extends this property to 3D rotations as well.

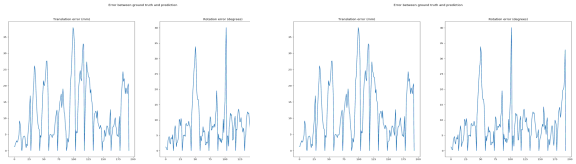

Fig. 4: The 3D Translational and Rotational error metrics, as they evolve in time, for the SoA and our proposed approaches.

5.2.2. The CookieJar model: an object with rotoreflective symmetry

For this special case, we also report our results without/with accounting for the object's symmetry axis in formulating our rotation loss. [2] achieves a translation error of 48.58 ± 38.23 mm and a rotation one of 35.43^0 $\pm$ 34.74^o (in terms of mean± std), while ours improves them to 11.63 ± 8.79 mm/$8.31^o \pm 6.76^o$. In fact, when we disentangle the rotation estimation and symmetries, we even reduce both metrics by $\%$. Namely, our approach outperforms [2] by $\%$ when we do not take symmetries into account and by $\%$ when we do. Qualitatively, in the second case we reduce the pose jitter that is inserted in the tracker's estimation due to the presence of the depth image source that describes the object's 3D shape.

6. CONCLUSION

In this work, we address challenges present in previous CNN pose tracking approaches by explicitly modelling them during network training. Clutter and occlusion handling auxiliary losses are added to the main tracking loss that accounts for the geometrical properties of both the Pose Space and the object 3D model. As a result, we reduce both pose error metrics by an average of 76% and gain an intuitive understanding of our artificial tracking mechanism.

7. REFERENCES

[1] Mathieu Garon and Jean-François Lalonde, "Deep 6-DOF Tracking," in *ISMAR*, 2017.

[2] Mathieu Garon, Denis Laurendeau and Jean-François Lalonde, "A Framework for Evaluating 6-DOF Object Trackers," in *ECCV*, 2018.

[3] Brégier Romain, Devernay Frédéric, Leyrit Laetitia and Crowley, James L., “Defining the Pose of Any 3D Rigid Object and an Associated Distance,” in *IJCV*, 2017.

[4] Yi Zhou and Connelly Barnes and Jingwan Lu and Jimei Yang and Hao Li, “On the Continuity of Rotation Representations in Neural Networks,” in *CVPR*, 2019.

[5] Alex Kendall and Yarin Gal and Roberto Cipolla, “Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics,” in *CVPR*, 2017.

[6] Hinterstoisser, Stefan and Lepetit, Vincent and Wohlhart, Paul and Konolige, Kurt, “On pre-trained image features and synthetic images for deep learning,” in *ECCV*, 2018.

[7] Mahendran, Siddharth and Ali, Haider and Vidal, René, “3d pose regression using convolutional neural networks,” in *ICCVW*, 2017.

[8] Sergey Zakharov and Ivan Shugurov and Slobodan Ilic, “DPOD: 6D Pose Object Detector and Refiner,” in *ICCV*, 2019.

[9] Hosseini Jafari, Omid and Mustikovela, Siva Karthik and Pertsch, Karl and Brachmann, Eric and Rother, Carsten, “iPose: Instance-Aware 6D Pose Estimation of Partly Occluded Objects,” in *CVPR*, 2019.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

[11] Forrest N. Iandola and Song Han and Matthew W. Moskewicz and Khalid Ashraf and William J. Dally and Kurt Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” in *NIPS*, 2016.

[12] Huang, Gao and Liu, Zhuang and van der Maaten, Laurens and Weinberger, Kilian Q , “Densely connected convolutional networks,” in *CVPR*, 2017.

[13] Chuong V. Nguyen and Shahram Izadi and David R. Lovell, “Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking,” in *IC3DIMPVT*, 2013.

[14] Jianxiong Xiao and Andrew Owens and Antonio Torralba, “SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels,” in *ICCV*, 2013.

[15] Clevert, Djork-Arné and Unterthiner, Thomas and Hochreiter, Sepp, “Fast and accurate deep network learning by exponential linear units (elus),” in *ISWC*, 2015.

[16] Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *ICCV*, 2015.

[17] Loshchilov, Ilya and Hutter, Frank, “Sgdr: Stochastic gradient descent with warm restarts,” *CoRR*, vol. abs/1804.10660, 2018.

[18] Srivastava, Nitish and Hinton, Geoffrey and Krizhevsky, Alex and Sutskever, Ilya and Salakhutdinov, Ruslan, “Dropout: a simple way to prevent neural networks from overfitting,” in *JMLR*, 2015.

[19] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, “Mask r-cnn,” in *ICCV*, 2017.

[21] Loshchilov, Ilya and Hutter, ”Frank Fixing weight decay regularization in adam,” , vol. abs/1301.3781, 2017.

[22] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang, “Deeptam: Deep tracking and mapping,” in *ECCV*, 2018.

[23] Li, Yi and Wang, Gu and Ji, Xiangyang and Xiang, Yu and Fox, Dieter, “Deepim: Deep iterative matching for 6d pose estimation,” *ECCV*, 2018.

[24] Laurens van der Maaten and Geoffrey E. Hinton, “Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again,” *JMLR*, vol. 9, 2008.

[25] Kehl, Wadim and Manhardt, Fabian and Tombari, Federico and Ilic, Slobodan and Navab, Nassir, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” *ICCV*,2017.

[26] Xiang, Yu and Schmidt, Tanner and Narayanan, Venkatraman and Fox, Dieter, “Generalized-icp”, : science and systems ,2009.

[27] Lepetit, Vincent and Moreno-Noguer, Francesc and Fua, Pascal, “Epnp: An accurate o (n) solution to the pnp problem,” *ICCV*,2009.