Supplemental Material:
Learning joint reconstruction of hands and manipulated objects

Yana Hasson[1,2] Gül Varol[1,2] Dimitrios Tzionas[3] Igor Kalevatykh[1,2]
Michael J. Black[3] Ivan Laptev[1,2] Cordelia Schmid[1]

[1]Inria, [2]Département d'informatique de l'ENS, CNRS, PSL Research University
[3]MPI for Intelligent Systems, Tübingen

Our main paper proposed a method for joint reconstruction of hands and objects. Below we present complementary analysis for hand-only reconstruction in Section A and object-only reconstruction in Section B. Section C presents implementation details.

A. Hand pose estimation

We first present an ablation study for the different losses we defined on the MANO hand model (Section A.1). Then, we study the latent hand representation (Section A.2). Finally, we validate our hand pose estimation branch and demonstrate its competitive performance compared to the state-of-the-art methods on a benchmark dataset (Section A.3).

A.1. Loss study on MANO

As explained in Section 3.1 of the main paper, we define three losses for the differentiable hand model while training our network: (i) vertex positions $\mathcal{L}_{V_{Hand}}$, (ii) joint positions $\mathcal{L}_J$, and (iii) shape regularization $\mathcal{L}_\beta$. The shape is only predicted in the presence of $\mathcal{L}_\beta$. In the absence of shape regularization, when only sparse keypoint supervision is provided, predicting β without regularizing it produces extreme deformations of the hand mesh, and we therefore fix β to the average hand shape.

Table A.1 summarizes the contribution of each of these losses. Note that the dense vertex supervision is available on our synthetic dataset ObMan, and not available on the real datasets FHB [5] and StereoHands [19].

We find that predicting β while regularizing it with $\mathcal{L}_\beta$ significantly improves the mean end-point-error on keypoints. On the synthetic dataset ObMan, we find that adding $\mathcal{L}_V$ yields a small additional improvement. We therefore use all three losses whenever dense vertex supervision is available, and $\mathcal{L}_J$ in conjunction with $\mathcal{L}_\beta$ when only keypoint supervision is provided.

	ObMan	FHB	StereoHands
$\mathcal{L}_J$	13.5	28.1	11.4
$\mathcal{L}_J + \mathcal{L}_\beta$	11.7	**26.5**	**10.0**
$\mathcal{L}_{V_{Hand}}$	14.0	-	-
$\mathcal{L}_{V_{Hand}} + \mathcal{L}_\beta$	12.0	-	-
$\mathcal{L}_{V_{Hand}} + \mathcal{L}_J + \mathcal{L}_\beta$	**11.6**	-	-

Table A.1: We report the mean end-point error (mm) to study different losses defined on MANO. We experiment with the loss on 3D vertices ($\mathcal{L}_{V_{Hand}}$), 3D joints ($\mathcal{L}_J$), and shape regularization ($\mathcal{L}_\beta$). We show the results of training and testing on our synthetic ObMan dataset, as well as the real datasets FHB [5] and StereoHands [19].

A.2. MANO pose representation

As described in Section 3.1 of the main paper, our hand branch outputs a 30-dimensional vector to represent the hand. These are the 30 first PCA components from the 45-dimensional full pose space. We experiment with different dimensionality for the latent hand representation and summarize our findings in Table A.2. While low-dimensionality fails to capture some poses present in the datasets, we do not observe improvements after increasing the dimensionality more than 30. Therefore, we use this value for all experiments in the main paper.

#PCA comps.	6	15	30	45
FHB	28.2	27.5	**26.5**	26.9
StereoHands	13.9	11.1	**10.0**	**10.0**
ObMan	23.4	13.3	11.6	**11.2**

Table A.2: We report the mean end-point error on error on multiple datasets to study the effect of the number of PCA hand pose components for the latent MANO representation.

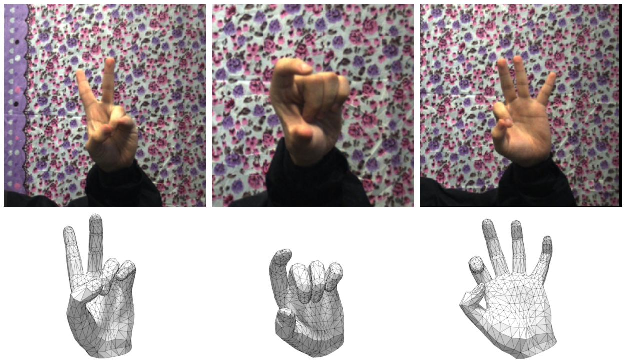


Figure A.1: Qualitative results on the test sequence of the StereoHands dataset.

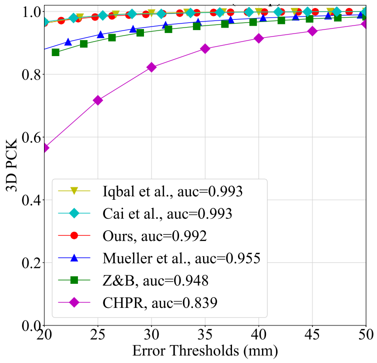


Figure A.2: We compare our root-relative 3D hand pose estimation on Stereohands to the state-of-the-art methods from Iqbal *et al.* [10], Cai *et al.* [1], Mueller *et al.* [15], Zimmermann and Brox [20], and CHPR [17].

A.3. Comparison with the state of the art

Using the MANO branch of the network, we can also estimate the hand pose for images in which the hands are not interacting with objects, and compare our results with previous methods. We train and test on the StereoHands dataset [19], and follow the evaluation protocol of [10, 15, 20] by training on 10 sequences from StereoHands and testing on the 2 remaining ones. For fair comparison, we add a palm joint to the MANO model by averaging the positions of two vertices on the front and back of the hand model at the level of the palm. Although the hand shape parameter β allows to capture the variability of hand shapes which occurs naturally in human populations, it does not account for the discrepancy between different joint conventions. To account for skeleton mismatch, we add a linear layer initialized to identity which maps from the MANO joints to the final joint annotations.

We report the area under the curve (auc) on the percentage of correct keypoints (PCK). Figure A.2 shows that our differentiable hand model is on par with the state of the art. Note that the StereoHands benchmark is close to saturation. In contrast to other methods [1, 10, 15, 17, 20] that only predicts sparse skeleton keypoints, our model produces a *dense* hand mesh. Figure A.1 presents some qualitative results from this dataset.

B. Object reconstruction

In the following, we validate our design choices for the object reconstruction branch. We experiment with object reconstruction (i) in the camera viewpoint (Section B.1) and (ii) with regularization losses (Section B.2).

B.1. Canonical versus camera view reconstruction

As explained in Section 3.2 of the main paper, we perform object reconstructions in the camera coordinate frame. To validate that AtlasNet [8] can successfully predict objects in camera view as well as in canonical view, we reproduce the training setting of the original paper [8]. We use the setting where 2500 points are sampled on a sphere and train on the rendered images from ShapeNet [2]. To obtain the rotated reference for the object, we apply the ground truth azimuth and elevation provided with the renderings so that the 3D ground truth matches the camera view. We use the original hyperparameters (Adam [12] with a learning rate of 0.001) and train both networks for 25 epochs. Both for supervision and evaluation metrics, we report the Chamfer distance $\mathcal{L}_{V_{Obj}} = \frac{1}{2}(\sum_p min_q \|p-q\|_2^2 + \sum_q min_p \|q-p\|_2^2)$ where q spans the predicted vertices and p spans points uniformly sampled on the surface of the ground truth object. We always sample the same number of points on the surface as there are vertices in the predicted mesh. We find that both numerically and qualitatively the performance is comparable for the two settings. Some reconstructed meshes in camera view are shown in Figure A.3. For better readability they also multiply the Chamfer loss by 1000. In order to provide results directly comparable with the original paper [8], we also report numbers with the same scaling in Table A.3. Table A.3 reports the Chamfer distances for their released model, our reimplementation in canonical

	Object error
Canonical view [8]	4.87
Canonical view (ours)	4.88
Camera view (ours)	4.88

Table A.3: Chamfer loss (×1000) for 2500 points in the canonical view and camera view show no degradation from predicting the camera view reconstruction. We compare our re-implementation to the results provided by [8] on their code page https://github.com/ThibaultGROUEIX/AtlasNet.

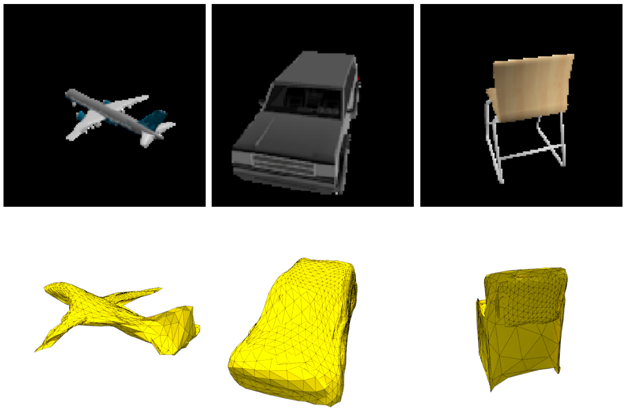


Figure A.3: Renderings from ShapeNet models and our corresponding reconstructions in camera view.

view, and our implementation in non-canonical view. We find that our implementation allows us to train a model with similar performances to the released model. We observe no numerical or qualitative loss in performance when predicting the camera view instead of the canonical one.

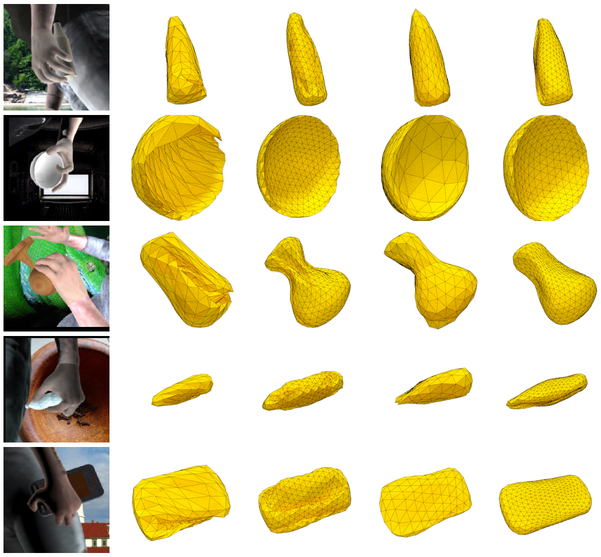
B.2. Object mesh regularization

We find that in the absence of explicit regularization on their quality, the predicted meshes can be very irregular. Sharp discontinuities in curvature occur in regions where the ground truth mesh is smooth, and the mesh triangles can be of very different dimensions. These shortcomings can be observed on all three reconstructions in Figure A.3. Following recent work on mesh estimation from image inputs [7, 11, 18], we introduce regularization terms on the object mesh.

Laplacian smoothness regularization ($\mathcal{L}_L$). In order to avoid unwanted discontinuities in the curvature of the mesh, we enforce a local prior of smoothness. We use the discrete Laplace-Beltrami operator to estimate the curvature at each mesh vertex position, as we have no prior on the final shape of the geometry, we compute the graph laplacian L on our mesh, which only takes into account adjacency between mesh vertices. Multiplying the laplacian L by the positions of the object vertices $\mathcal{V}_{Obj}$ produces vectors which have the same direction as the vertex normals and their norm proportional to the curvature. Minimizing the norm of these vector therefore minimizes the curvature. We minimize the mean curvature over all vertices in order to encourage smoothness on the mesh.

Laplacian edge length regularization ($\mathcal{L}_E$). $\mathcal{L}_E$ penalizes configurations in which the edges of the mesh have different lengths. The edge regularization is defined as:

$$\mathcal{L}_E = \frac{1}{|\mathcal{E}_L|} \sum_{l \in \mathcal{E}_L} |l^2 - \mu(\mathcal{E}_L^2)|, \qquad (1)$$



	No reg.	$\mathcal{L}_E$	$\mathcal{L}_L$	$\mathcal{L}_E + \mathcal{L}_L$
Object error	0.0246	0.0286	0.0258	0.0292

Figure A.4: We show the benefits from each term of the regularization. Using both the $\mathcal{L}_E$ and $\mathcal{L}_L$ in conjunction improves the visual quality of the predicted triangulation while preserving the shape of the object.

where $\mathcal{E}_L$ is the set of edge lengths, defined as the L2 norms of the edges, and $\mu(\mathcal{E}_L^2)$ is the average of the square of edge lengths.

To evaluate the effect of the two regularization terms we train four different models. We train a model without any regularization, two models for which only one of the two regularization terms are active, and finally a model for which the two regularization terms are applied simultaneously. Each of these models is trained for 200 epochs.

Figure A.4 shows the qualitative benefits of each term. While edge regularization $\mathcal{L}_E$ alone already significantly improves the quality of the predicted mesh, note that unwanted bendings of the mesh still occur, for instance in the last row for the cellphone reconstruction. Adding the laplacian smoothness $\mathcal{L}_L$ resolves these irregularities. However, adding each regularization term negatively affects the final reconstruction score. Particularly we observe that introducing edge regularization increases the Chamfer loss by 22% while significantly improving the perceptual quality of the predicted mesh. Introducing the regularization terms contributes to the coarseness of the object reconstructions, as can be observed on the third row, where sharp curvatures of the object in the input image are not captured in the reconstruction.

C. Implementation details

We give implementation details on our training procedure (Section C.1) and our automatic grasp generation (Section C.2).

C.1. Training details

For all our experiments, we use the Adam optimizer [12]. As we observe instabilities in validation curves when training on synthetic datasets, we freeze the batch normalization layers. This fixes their weights to the original values from the ImageNet [16] pre-trained ResNet18 [9].

For the final model trained on ObMan, we first train the (normalized) object branch using $\mathcal{L}^n_{Object}$ for 250 epochs, we start with a learning rate of 10^{-4} and decrease it to 10^{-5} at epoch 200. We then freeze the object encoder and the AtlasNet decoder, as explained in Section 3.2 of the main paper. We further train the full network with $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$ for 350 additional epochs, decreasing the learning rate from 10^{-4} to 10^{-5} after the first 200 epochs.

When fine-tuning from our main model trained on synthetic data to smaller real datasets, we unfreeze the object reconstruction branch.

For the FHB$_c$ dataset, we train all the parts of the network simultaneously with the supervision $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$ for 400 epochs, decreasing the learning rate from 10^{-4} to 10^{-5} at epoch 300.

When fine-tuning our models with the additional contact loss, $\mathcal{L}_{Hand} + \mathcal{L}_{Object} + \mu_C \mathcal{L}_{Contact}$, we use a learning rate of 10^{-5}. We additionally set the momentum of the Adam optimizer [12] to zero, as we find that momentum affects negatively the training stability when we include the contact loss.

In all experiments, we keep the relative weights between different losses as provided in the main paper and normalize them so that the sum of all the weights equals 1.

C.2. Heuristic metric for sorting GraspIt grasps

We use GraspIt [14] to generate grasps for the ShapeNet object models. GraspIt generates a large variety of grasps by exploring different initial hand poses. However, some initializations do not produce good grasps. Similarly to [6] we filter the grasps in a post-processing step in order to retain grasps of good quality according to a heuristic metric we engineer for this purpose.

For each grasp, GraspIt provides two grasp quality metrics ε and v [4]. Each grasp produced by GraspIt [14] defines contact points between the hand and the object. Assuming rigid contacts with friction, we can compute the space of wrenches which can be resisted by the grasp: the grasp wrench space (GWS). This space is normalized with relation to the scale of the object, defined as the maximum radius of the object, centered at its center of mass. The grasp is suitable for any task that involves external wrenches that lie within the GWS. v is the volume of the 6-dimensional GWS, which quantifies the range of wrenches the grasp can resist. The GWS can further be characterized by the radius ε of the largest ball which is centered at the origin and inscribed in the grasp wrench space. ε is the maximal wrench norm that can be balanced by the contacts for external wrenches applied coming from arbitrary directions. ε belongs to $[0, 1]$ in the scale-normalized GWS, and higher values are associated with a higher robustness to external wrenches.

We require a single value to reflect the quality of the grasp in order to sort different grasps. We use the norm of the $[\varepsilon, v]$ vector in our heuristic measure of grasp quality. We find that in the grasps produced by GraspIt, power grasps, as defined by [3] in which larger surfaces of the hand and the object are in contact, are rarely produced. To allow for a larger proportion of power grasps, we use a multiplier γ_{palm} which we empirically set to 1 if the palm is not in contact and 3 otherwise. We further favor grasps in which a large number of phalanges are in contact with the object by weighting the final grasp score using N_p, the number of phalanges in contact with the object, which is computed by the software.

The final grasp quality score G is defined as:

$$G = \gamma_{palm} \sqrt{N_p} \|\varepsilon, v\|_2. \qquad (2)$$

We find that keeping the two best grasps for each object produces both diverse grasps and grasps of good quality.

D. Qualitative results on CORe50 dataset

We present additional qualitative results on the CORe50 [13] dataset. We present a variety of diverse input images from CORe50 in Figure A.5 alongside the predictions of our final model trained solely on ObMan.

The first row presents results on various shapes of light bulbs. Note that this category is not included in the synthetic object models of ObMan. Our model can therefore generalize across object categories. The last column shows some reconstructions of mugs, showcasing the topological limitations of the sphere baseline of AtlasNet which cannot, by construction, capture handles.

However, we observe that the object shapes are often coarse, and that fine details such as phone antennas are not reconstructed. We also observe errors in the relative position between the object and the hand, which is biased towards predicting the object's centroid in the palmar region of the hand, see Figure A.5, fourth column. As hard constraints on collision are not imposed, hand-object interpenetration occurs in some configurations, for instance in the top-right example. In the bottom-left example we present a failure case where the hand pose violates anatomical constraints. Note that while our model predicts hand pose in

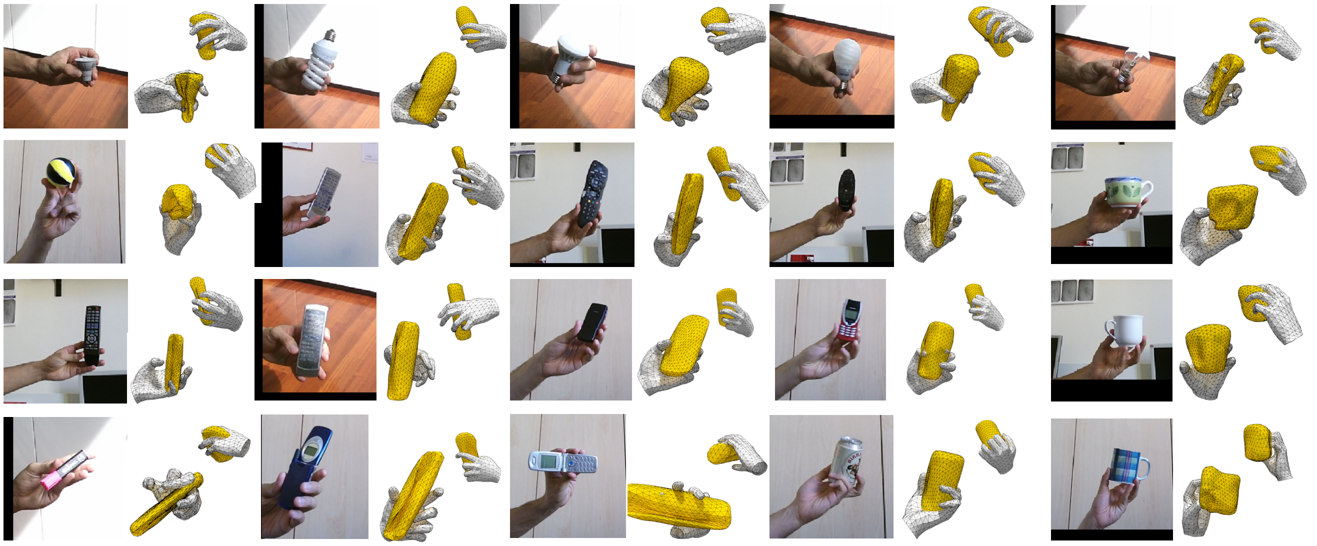


Figure A.5: Qualitative results on CORe50 dataset. We present additional hand-object reconstructions for a variety of object categories and object instances, spanning various hand poses and object shapes.

a low-dimensional space, which implicitly regularizes hand poses, anatomical validity is not guaranteed.

References

[1] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*, 2018. 2

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, 2016. 2

[3] T. Feix, J. Romero, H.-B. Schmiedmayer, A. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *Human-Machine Systems, IEEE Transactions on*, 2016. 4

[4] C. Ferrari and J. F. Canny. Planning optimal grasps. In *ICRA*, 1992. 4

[5] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 1

[6] C. Goldfeder, M. T. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *ICRA*, 2009. 4

[7] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. 3D-CODED : 3D correspondences by deep deformation. In *ECCV*, 2018. 3

[8] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 4

[10] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 2

[11] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 3

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 2, 4

[13] V. Lomonaco and D. Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, Proceedings of Machine Learning Research, 2017. 4

[14] A. T. Miller and P. K. Allen. Graspit! A versatile simulator for robotic grasping. *Robotics Automation Magazine, IEEE*, 11:110 – 122, 2004. 4

[15] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018. 2

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

[17] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. 2015. 2

[18] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, 2018. 3

[19] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3D hand pose tracking and estimation using stereo matching. *arXiv:1610.07214*, 2016. 1, 2

[20] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single rgb images. In *ICCV*, 2017. 2