



Reconstructing hands and manipulated objects from images and videos

Yana Hasson

► To cite this version:

Yana Hasson. Reconstructing hands and manipulated objects from images and videos. Computer Vision and Pattern Recognition [cs.CV]. Inria, 2021. English. NNT: . tel-03616841

HAL Id: tel-03616841

<https://hal.science/tel-03616841>

Submitted on 23 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

**Reconstructing hands and manipulated objects
from images and videos**

Soutenue par

Yana HASSON

Le 13 Octobre 2021

Ecole doctorale n° 386

**Sciences Mathématiques
de Paris Centre**

Spécialité

Informatique

Composition du jury :

Patrick, PÉREZ
Valeo

Président du jury

Jürgen, GALL
Universität Bonn

Rapporteur

Vincent, LEPETIT
ENPC ParisTech

Rapporteur

Marc, POLLEFEYS
ETH Zurich

Examineur

Ivan, LAPTEV
Inria

Directeur de thèse

Cordelia, SCHMID
Inria

Directrice de thèse



List of Abbreviations

ADL Activities of Daily Living. , 2, 5

API Application Programming Interface. 4

AR Augmented Reality. , 4, 5, 40

CAD Computer-aided Design. 23, 27, 29, 30, 123

CNN Convolutional Neural Network. , 10, 12, 21, 23, 27, 28, 29, 30, 31, 33, 35, 36, 37, 41, 51, 52, 55, 56, 58, 67, 123

DNN Deep Neural Network. 23, 24

DOF Degree of Freedom. 29, 31, 35

fps frames per second. 29

GAN Generative Adversarial Network. 41, 59

GCN Graph Convolutional Network. 24, 39, 51

GWS Grasp Wrench Space. 58, 62

HMD Head Mounted Display. 4

HOG Histogram of Oriented Gradients. 35, 40, 41

ICP Iterative Closest Point. 43

LBS Linear Blend Skinning. 35, 36, 43, 45

LfD Learning from Demonstration. 5

LSTM Long Short-Term Memory. 51

MLP Multi Layer Perceptron. 24

MoCap Motion Capture. 44, 49, 50, 58

NMR Neural Mesh Renderer. 25, 31

PCA Principal Component Analysis. 37, 38

PSO Particle Swarm Optimization. 37, 43, 44, 49

RGB Red Green Blue. 7, 10, 53, 55

SDF Signed Distance Function. 19, 22

SDM Signed Distance Map. 16, 17, 19, 20, 48

SOTA State of the art. 99

SVM Support-Vector Machine. 23

SVR Single View Reconstruction. , 21, 22, 24, 27

VAE Variational Auto-Encoder. 59

Abstract

Modeling hand-object manipulations is essential for understanding how humans interact with their environment. Recent efforts to recover 3D information from RGB images have been directed towards fully-supervised methods which require large amounts of labeled training samples. However, collecting 3D ground-truth data for hand-object interactions is costly, tedious, and error-prone. In this thesis, we propose several contributions to overcome this challenge.

First, we propose a fully automatic method to generate synthetic data with hand-object interactions for training. We generate ObMan, a synthetic dataset with automatically generated labels, and use it to train a deep convolutional neural network to reconstruct the observed object and the hand pose from a single RGB frame. We present an end-to-end learnable model that exploits a novel contact loss to favor physically plausible hand-object constellations. We investigate the domain gap and validate that our synthesized training data allows our model to reconstruct hand-object interactions from real images, provided the captured grasps are similar to the ones in the synthetic images.

While costly, curating annotations from real images allows to obtain samples from the distribution of natural hand-object interactions. Next, we investigate a strategy to make the most of manual annotation efforts: we propose to leverage the temporal context in videos when sparse annotations are available. In a learnable framework which jointly reconstructs hands and objects in 3D by inferring the poses of known models, we leverage photometric consistency across time. Given our estimated reconstructions, we differentially render the optical flow between pairs of images and use it to warp one frame to another. We then apply a self-supervised photometric loss that relies on the visual consistency between nearby images. We display competitive results for 3D hand-object reconstruction benchmarks and demonstrate that our approach allows to improve the pose estimation accuracy by leveraging information from neighboring frames in low-data regimes.

Finally, we explore automatic annotation of real RGB data by proposing a learning-free fitting approach for hand-object reconstruction. We rely on 2D cues obtained with common learnt methods for detection, hand pose estimation and instance segmentation and integrate hand-object interaction priors. We evaluate our approach and show that it can be applied to datasets with varying levels of complexity. Our method can seamlessly handle two-hand object interactions and can provide noisy pseudo-labels for learning-based approaches.

In summary, our contributions are the following: (i) we generate synthetic data for hand-object grasps that allows training CNNs for joint hand-object reconstruction, (ii) we propose a strategy to leverage the temporal context in videos when sparse annotations are provided, (iii) we propose to recover hand-object interactions for short video clips by fitting models to noisy predictions from learnt models.

Résumé

Modéliser la manipulation d'objets est essentiel à la compréhension des interactions entre l'homme et son environnement. Les efforts pour reconstituer l'information 3D à partir d'images RGB se sont récemment orientés vers les méthodes d'apprentissage supervisés, qui requièrent de nombreux exemples annotés durant l'entraînement. Cependant, la collecte d'annotations précises pour des images de manipulation est laborieuse, coûteuse, et propice aux erreurs. Cette thèse propose plusieurs contributions pour pallier ces difficultés.

Premièrement, nous proposons une méthode pour générer automatiquement des données synthétiques d'interactions mains-objets. Nous générons un set de données synthétiques: ObMan, que nous utilisons pour entraîner un réseau de neurones convolutif profond pour la reconstruction 3D à partir d'une unique image RGB. Nous présentons une méthode d'apprentissage différentiable qui favorise des reconstructions physiquement plausibles à l'aide d'une nouvelle pénalisation. Nos données synthétiques permettent au modèle de reconstruire des interactions à partir d'images réelles, à condition que les configurations observées soient proches des manipulations représentées par les images synthétiques.

Bien que coûteuse, l'annotation d'images réelles permet d'obtenir des exemples d'interactions mains-objets qui surviennent naturellement lors de la manipulation. Nous proposons de valoriser des annotations obtenues manuellement en utilisant le contexte temporel pour des vidéos annotées sporadiquement. Nous déployons une architecture différentiable qui permet de reconstruire les interactions 3D en estimant la pose de la main et d'un objet manipulé dont le modèle est supposé connu, et proposons d'utiliser la consistance photométrique au cours du temps comme signal d'entraînement. Étant données nos prédictions, nous estimons le flux optique entre deux images à l'aide d'un rendu différentiable, et utilisons celui-ci pour transposer les déformations dues aux mouvements d'une des images vers l'autre. Nous appliquons ensuite une pénalité photométrique auto-supervisée basée sur la cohérence visuelle entre images proches. Notre méthode produit des résultats compétitifs pour des jeux de données de référence en reconstruction main-objet. Nous démontrons que notre approche permet d'améliorer la précision de l'estimation de pose en mettant à profit les informations des images voisines lorsque la densité d'annotation est faible.

Enfin, nous étudions l'annotation automatique de données RGB réelles en proposant une approche d'optimisation sans apprentissage pour la reconstruction de manipulations d'objets. Nous nous appuyons sur des prédictions produites par des modèles d'apprentissage établis pour la détection, l'estimation de la pose de la main et la segmentation d'instance, et intégrons des heuristiques régulant les interactions mains-objets. Nous évaluons notre approche et montrons qu'elle peut être appliquée à des vidéos présentant différents niveaux de complexité. Notre méthode peut modéliser les manipulations entre plusieurs mains et un objet et fournir des annotations bruitées pour des méthodes basées sur l'apprentissage.

En résumé, nos contributions sont les suivantes : (i) nous générons des données synthétiques pour la saisie d'objets qui permettent d'entraîner des réseaux convolutif profonds pour la reconstruction jointe d'une main et d'un objet, (ii) nous proposons une stratégie pour utiliser le contexte temporel des vidéos lorsque la densité temporelle d'annotation est faible, (iii) nous proposons d'estimer les interactions main-objet pour de courts clips vidéo en mettant en correspondance les poses 3D de mains et d'objets avec les prédictions de modèles entraînés.

Contents

1	Introduction	1
1.1	Applications	4
1.1.1	Applications to AR.	4
1.1.2	Applications to robotics.	5
1.2	Challenges	6
1.2.1	Representing manipulated objects.	6
1.2.2	Cluttered scenes	7
1.2.3	Access to limited information.	8
1.2.4	Limited 3D annotations.	9
1.3	Goal	10
1.4	Contributions	10
1.4.1	Publications	11
1.4.2	Software & dataset contributions	12
1.5	Outline	13
2	Related work	15
2.1	Object modeling	15
2.1.1	Object representation	16
2.1.2	Object shape estimation	19
2.1.3	Rigid pose estimation	27
2.2	Hand modeling	34
2.2.1	Articulated pose estimation	35
2.2.2	Keypoint regression	35
2.2.3	Hand pose and shape estimation	36
2.3	Joint hand-object reconstruction	40
2.3.1	Objects as occluders.	40
2.3.2	Joint optimization	41
2.3.3	Hand-object 3D annotations.	50
2.3.4	Hand-object reconstruction from a single RGB frame	51

3	Learning joint hand-object reconstruction from synthetic grasps	55
3.1	Related work	56
3.1.1	Synthetic data rendering	56
3.1.2	Automatic grasp generation	58
3.2	Generating hand-object interactions	59
3.2.1	Automatic grasp generation	59
3.2.2	Grasp rendering	63
3.3	Learning grasp reconstruction	67
3.3.1	Regressing hand parameters	67
3.3.2	Object mesh estimation	69
3.3.3	Contact loss	71
3.3.4	Implementation details	73
3.4	Experiments	73
3.4.1	Evaluation metrics	74
3.4.2	Datasets	74
3.4.3	Hand pose estimation	76
3.4.4	Object reconstruction	78
3.4.5	Effect of occlusions	80
3.4.6	Effect of contact loss	82
3.4.7	Synthetic to real transfer	83
3.4.8	Qualitative results on CORE50	86
3.5	Conclusions	88
4	Learning from sparse annotations	89
4.1	Related work	90
4.1.1	Learning with temporal constraints	90
4.1.2	Learning with photometric consistency	90
4.2	Learning to grasp known objects with sparse temporal supervision	91
4.2.1	Temporal supervision from sparse grasp supervision	91
4.2.2	Learning to grasp known objects	93
4.2.3	Dense 3D Hand-Object Reconstruction	97
4.3	Experiments	97
4.3.1	Datasets	97
4.3.2	Evaluation Metrics	98
4.3.3	Experimental Results	99
4.3.4	Skeleton adaptation	105
4.3.5	Runtime.	105
4.4	Conclusions	105

5	Joint hand-object fitting	107
5.1	Related work	108
5.1.1	Annotating 3D objects in real images.	108
5.1.2	Joint fitting to RGB frames.	109
5.1.3	Temporal constraints for motion modeling.	109
5.2	Fitting hand-object interactions in RGB images	110
5.2.1	Obtaining 2D hand-object evidence	110
5.2.2	Independent pose initialization	111
5.2.3	Joint fitting	111
5.3	Learning from noisy data	114
5.4	Experiments	114
5.4.1	Metrics	114
5.4.2	Datasets	115
5.4.3	Contribution of error terms in fitting	116
5.4.4	Sensitivity to estimated 2D evidence	117
5.4.5	State-of-the-art comparison	119
5.4.6	In-the-wild 3D hand-object pose estimation	121
5.5	Conclusions	122
6	Discussion	123
6.1	Summary of contributions	123
6.2	Perspectives	124
6.2.1	Importance and limitations of existing datasets.	124
6.2.2	Hybrid annotation methods.	124
6.2.3	Learning from noisy annotations.	125
6.2.4	Modeling object state changes	125
6.2.5	Discovering statistical object affordances	126

List of Figures

1-1	Scenes can typically be visually factorized at the pixel level into active agents, most often humans, objects and the surrounding static environment. We illustrate this factorization on an example image from the Epic-Kitchens 2018 dataset Damen et al. (2018).	1
1-2	Example of hand-object interactions which occur in a kitchen environment during food and drink preparation from the Epic-Kitchens 2018 dataset Damen et al. (2018). Object manipulation plays a crucial roles in activities of daily living (ADL).	2
1-3	Various examples of object manipulation. People typically use their hands everyday to accomplish first-necessity tasks as well as for leisure and social activities. Images by Debora Alves, Sabine Ponce Joshua Woroniecki, Sabine van Erp, Renate Köpel, from pixabay.	3
1-4	Famous examples of object manipulation.	3
1-5	Accurate modeling of unconstrained hand-object interactions could help provide guidance during manipulation using AR devices (see Section 1.1.1) or transfer useful skills to robots (see Section 1.1.2).	5
1-6	Everyday objects vary in shape. Even for a single seemingly constrained category such as mugs, a wide variety of shapes are observed in practice, see the first row. The variety of object shapes is reflected in CAD databases such as ShapeNet Chang et al. (2015) which source their models from the web, which we illustrate in the second row.	6
1-7	Different representations capture 3D information of material objects. 3D representations are discussed in more details in Section 2.1.1	7
1-8	. Images and videos displaying natural hand-object interactions present many challenges. Note that difficulties often accumulate in a single image. The top right image for instance suffers from extreme illumination in addition to background clutter and occlusions.	8

1-9	Recent efforts have aimed to scale data annotation for real datasets depicting hand-object interactions. However, the total number of annotated objects and sequences is still limited.	9
2-1	Different 3D representations allow to capture and encode a given 3D shape, such as this mug from the ShapeNet Chang et al. (2015) dataset. Each representation incurs a different trade-off in terms of flexibility, modeling capacity and storage costs, determining its adequacy for a given task. 3D representations are discussed in more details in Section 2.1.1	17
2-2	One of the first attempts to automatically recover the shape of 3D objects from 2D images by Roberts (1963). Images from Roberts (1963)	20
2-3	Recent learning-based methods demonstrate SVR results for real and synthetic images by recovering the parameters for a variable number of shape primitives.	22
2-4	AtlasNet Groueix et al. (2018b) learns to deform a template by mapping its surface points to surface point coordinates of the reconstructed object. . . .	24
2-5	Differentiable rendering allows to deform a mesh template so that it matches a given silhouette (2-5a. Here we display optimization results where we fit a spherical mesh to a target shape silhouette by optimizing each vertex location independently using Pytorch3D. While the outline imposes valid constraints on the object shape, the problem is underconstrained in the case of single-view silhouette optimization, resulting in implausible shapes (2-5b). Additional regularization constraints, in this case a weighted sum of Laplacian, normal smoothness and edge length regularization (with weights 1, 0.01 and 1 respectively, as per the Pytorch3D Ravi et al. (2020) tutorial) provide an improvement but fail to capture semantic priors (2-5c).	26
2-6	Differentiable rendering allows to optimize the translation and rotation of a target mesh to match silhouette or RGB evidence. Above, we present successful (2-6a) and unsuccessful (2-6b) examples where the object pose is optimized using the Pytorch3D Ravi et al. (2020) tutorial by matching a reference and differentially silhouette mask starting from two different initial poses.	32
2-7	The MANO Romero et al. (2017) parametric hand model.	38
2-8	Several hand-object datasets have been proposed to support methods which focus on estimating hand poses and model the occlusions generated by objects.	42

2-9	Examples of input frames and reconstructed hand-object configurations for model-based tracking methods. All images are reproduced from the original papers.	43
2-10	While Oikonomidis et al. (2011b) model collisions by associating collision primitives to their hand model, Ballan et al. (2012) take into account the dense mesh surface and speed-up computations by computing local triangle-triangle intersections. Images reproduced from the original papers (Ballan et al. (2012); Oikonomidis et al. (2011b))	47
3-1	We select objects from 8 graspable object categories from the ShapeNet Chang et al. (2015) database.	60
3-2	We select objects from the ShapeNet Chang et al. (2015) database. We present here random examples of objects from the vase category which we use for our ObMan dataset in the ShapeNet model exploration interface https://shapenet.org/model-querier . Objects in ShapeNet can be composed of an arbitrary number of parts and present different topologies.	61
3-3	We present different grasps generated by GraspIt Miller and Allen (2004) for a single object CAD model as described in Section 3.2.1 in the first row, and grasps for 4 different ShapeNet Chang et al. (2015) models in the second row.	61
3-4	We render a full-body posed human model grasping an object with realistic textures obtained by combining body and hand textures	63
3-5	Textures from full body scans typically display missing and erroneous color values in the hand region. We use high-resolution hand scans color-aligned with the person’s face skin to inpaint the target hand region (see lower right of each body texture image, outlined in red before and in green after inpainting) as described in Section 3.2.2.	64
3-6	ObMan: large-scale synthetic dataset of hand-object interactions. We pose the MANO hand model Romero et al. (2017) to grasp a given object mesh using GraspIt Miller and Allen (2004), see Section 3.2.1. The scenes are rendered with variation in texture, lighting, and background, as described in Section 3.2.2.	64
3-7	We render object-only and hand-only images for each sample in the ObMan dataset along with depth maps for each of the hand-only object-only and joint configuration, which we store in different color channels as an image along with minimum and maximum depth values for efficiency.	65

3-8	Our ObMan dataset provides <i>synthetic</i> images with pixel-accurate segmentation maps, 3D hand joints as well as hand and object meshes in camera coordinates.	65
3-9	Our model predicts the hand and object meshes in a single forward pass in an end-to-end framework. The repulsion loss \mathcal{L}_R penalizes interpenetration while the attraction loss \mathcal{L}_A encourages the contact regions to be in contact with the object.	67
3-10	Left: Estimated contact regions from ObMan. We find that points that are often involved in contacts can be clustered into 6 regions on the palmar surface of the hand. Right: Generic shape of the penalization function emphasizing the role of the characteristic distances.	72
3-11	Qualitative results on the test sequence of the StereoHands dataset.	77
3-12	We compare our root-relative 3D hand pose estimation on Stereohands to the state-of-the-art methods from Iqbal et al. (2018), Cai et al. (2018), Mueller et al. (2018), Zimmermann and Brox (2017), and CHPR Sun et al. (2015).	78
3-13	Renderings from ShapeNet models and our corresponding reconstructions in camera view.	79
3-14	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.	81
3-15	Qualitative comparison between <i>with</i> (bottom) and <i>without</i> (top) contact on FPHAB _C . Note the improved contact and reduced penetration, highlighted with red regions, with our contact loss.	82
3-16	We examine the relative importance between the contact terms on the grasp quality metrics. Introducing a well-balanced contact loss improves upon the baseline on both max penetration and simulation displacement.	84
3-17	We compare training on FPHAB only (Real) and pre-training on synthetic, followed by fine-tuning on FPHAB (Synth2Real). As the amount of real data decreases, the benefit of pre-training increases. For both the object and the hand reconstruction, synthetic pre-training is critical in low-data regimes.	85
3-18	We compare the effect of training with and without fine-tuning on variants of our synthetic dataset on HIC. We illustrate each dataset (a, b, c, d) with an image sample, see text for definitions. Synthetic pre-training, whether or not the target distribution is matched, is always beneficial.	85

3-19	Qualitative results on CORE50. Our model, trained only on synthetic data, shows robustness to various hand poses, objects and scenes. Global hand pose and object outline are well estimated while fine details are missed. We present failure cases in the red box. Note that this model is trained on synthetic ObMan images only.	86
3-20	Selected 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. Each image shows manipulation of a different object model.	86
3-21	To show the typical performance of our model on the CORE50 dataset Lomonaco and Maltoni (2017), We display the outputs of our method on 25 randomly sampled frames from this dataset. Note that the images are randomly drawn from the <i>subset</i> of CORE50 which we annotated with hand side and hand-object region of interest.	87
4-1	Photometric consistency loss. Given an annotated frame, t_{ref} , and an unannotated one, t_{ref+k} , we reconstruct hand and object 3D pose at t_{ref+k} leveraging a self-supervised loss. We differentiably render the optical flow between ground-truth hand-object vertices at t_{ref} and estimated ones. Then, we use this flow to warp frame t_{ref+k} into t_{ref} , and enforce consistency in pixel space between warped and real image.	91
4-2	Architecture of the single-frame hand-object reconstruction network. Our network assumes that the object CAD model is available and regresses pose parameters for the hand and object as well as MANO hand shape parameters.	94
4-3	Qualitative results on the FPHAB dataset. We visualize the reconstructed meshes reprojected on the image as well as a rotated view. When training on the full dataset, we obtain reconstructions which accurately capture the hand-object interaction. In the sparsely supervised setting, we qualitatively observe that photometric consistency allows to recover more accurate hand and object poses. Failure cases occur in the presence of important motion blur and large occlusions of the hand or the object by the subject's arm. . .	98
4-4	Evaluation of our baseline for hand-object pose estimation on the early release of the HO-3D Hampali et al. (2019) dataset. We report the PCK for 2D joint mean-end-point error for hands, and the mean 2D reprojection error for objects.	101
4-5	Effect of using photometric-consistency self-supervision when only a fraction of frames are fully annotated on HO-3D. We report average values and standard deviations over 5 different runs.	101

4-6	We observe consistent quantitative improvements from the photometric consistency loss as the percentage of fully supervised frames decreases below 10% for both hands and objects.	102
4-7	Progressive pose refinement over training samples, even in the presence of large motion and inaccurate initialization. In extreme cases (last row), the model cannot recover.	103
4-8	Predicted reconstructions for images from HO-3D. While rotation errors around axis parallel to the camera plane are not corrected and are sometimes even introduced by the photometric consistency loss, we observe qualitative improvement in the 2D reprojection of the predicted meshes on the image plane.	104
4-9	Predicted shape deformations in the (a) absence and (b) presence of the skeleton adaptation layer on the FPHAB dataset.	105
5-1	Joint hand-object fitting: We independently initialize the hand and object poses based on 2D detections and segmentations. We refine this configuration with interaction-based constraints to obtain our final joint fitting.	110
5-2	Effect of error terms: Qualitative analysis showing the effects of the various error terms for the hand-object reconstruction accuracy on the HO-3D dataset. We highlight visual evidence of local corrections attributed to the local interaction from Chapter 3 and collision Jiang et al. (2020) terms. . . .	117
5-3	Sensitivity to 2D detections: Dependence of our 3D reconstruction on the accuracy of the 2D evidence by running our method with ground truth (GT) hand and object detections and ground truth object masks for the HO-3D dataset Hampali et al. (2020).	118
5-4	In-the-wild reconstructions: Our results on natural hand-object manipulations of the Epic-Kitchens dataset Damen et al. (2018). We present several success and failures of our method on the challenging Epic-Kitchens dataset. We highlight typical failure modes for our method, in particular, object orientation errors resulting from depth ambiguity. We observe that our fitting method recovers plausible interactions across different object categories and hand-object configurations.	120
5-5	Comparison with Chapter 3: Qualitative comparison of our fits to Chapter’s 3 ObMan-trained model estimations on the Core50 dataset. While our model requires an approximate mesh to be provided, it generalizes to objects of arbitrary topology.	121

6-1	Recent methods which model hand-object interactions can only model a restricted subset of objects, and typically do not model object state changes. Addressing these dynamic scenes which are beyond the scope of existing reconstruction methods requires carefully designing appropriate 3D scene representations.	126
6-2	Reconstructing hand-object interactions from a large number of demonstration videos could allow to capture statistical object affordances. Real demonstrations could be used to extract plausible agent motions which could guide exploration in simulated environments such as SAPIEN Xiang et al. (2020) or iGibson Shen et al. (2020). The figure is composed using illustrations from the SAPIEN dataset (Xiang et al. (2020)) and frames from YouTube videos describing microwave usage.	128

List of Tables

2.1	SVR results for recent CNN-based methods. 3D-R2N2 Choy et al. (2016) can reconstruct arbitrary objects but with a limited resolution. Pixel2Mesh Wang et al. (2018) produces detailed results but is restricted to a simple topology as it deforms a sphere template. AtlasNet Groueix et al. (2018b) reconstructs complex topologies but results in non-watertight meshes. Occupancy prediction Mescheder et al. (2019) results in precise watertight meshes but requires post-processing to extract the surface. Images courtesy of Mescheder et al. (2019).	27
3.1	Dataset details for train/test splits.	75
3.2	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}_β). We show the results of training and testing on our synthetic ObMan dataset, as well as the real datasets FPHAB Garcia-Hernando et al. (2018) and StereoHands Zhang et al. (2016).	76
3.3	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.	77
3.4	Chamfer loss ($\times 1000$) 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 Groueix et al. (2018b) on their code page https://github.com/ThibaultGROUEIX/AtlasNet .	79
3.5	We first show that training with occlusions is important when targeting images of hand-object interactions.	82
3.6	We experiment with each term of the contact loss. Attraction (\mathcal{L}_A) encourages contacts between close points while repulsion (\mathcal{L}_R) penalizes interpenetration. λ_R is the repulsion weight, balancing the contribution of the two terms.	83

4.1	Architecture of the Hand and Object parameter regression branches. We use fully connected linear layers to regress pose and shape parameters from the 512-dimensional features.	94
4.2	Comparison to state-of-the-art method of Tekin et al. (2019) on FPHAB Garcia-Hernando et al. (2018), errors are reported in mm.	99
4.3	We compare training for hand and object pose estimation jointly and separately on FPHAB Garcia-Hernando et al. (2018) and find that the encoder can be shared at a minor performance cost in hand and object pose accuracy.	100
4.4	On the FHPAB dataset, for which the skeleton is substantially different from the MANO one, we show that adding a skeleton adaptation layer allows us to outperform our results from Chapter 3, while additionally predicting the global translation of the hand.	104
5.1	Contribution of error terms: We show benefits of the <i>joint</i> modeling for hand-object interactions by the increased reconstruction accuracy when compared to independent hand and object composition on the HO-3D Hampali et al. (2020) dataset. Our smoothness and interaction terms impose additional constraints which improve the final hand-object pose reconstructions.	116
5.2	Results on Core50: Interaction errors for hand-object fits obtained on the Core50 dataset. We observe significantly improved contact accuracy with joint fitting over independent fits at the expense of a minor cost of a 0.6mm increase in penetration.	118
5.3	State-of-the-art-comparison: We compare the hand performance of the single-view baseline from Chapter 4 to previously reported methods on hand metrics. Note that the reported results for Hampali et al. (2020) and Chapter 3 are for methods which output hand meshes only, Chapter 4 and the method presented in this chapter predict the hand-object meshes jointly. All methods are trained only on the real images from the HO-3D training split and evaluated on the official test split through an online submission ¹ . .	119
5.4	Unseen objects: Vertex errors (cm) for estimated hand and object meshes. Compared to Chapter 4, our method performs similarly across seen and <i>unseen</i> objects and sees further benefits from test-time training.	120

Chapter 1

Introduction

People perceive their environment through sensory organs, which provide partial cues on the structure of their surroundings. Among the different senses, it is known that vision plays a central role. Vision is the most studied sensory modality (Hutmacher (2019); Sternberg and Sternberg (2017)) and humans rely on vision as one of their "primary source of objective data about the world" Sweetser (1990).

Recent development of affordable image sensors, storage devices and sharing platforms has led to an explosion of accessible image and video data. With more than 5 billion internet users at the end of 2020 according to Internet World Stats (2020), large databases of digital data are accessible to most of the population. Computer vision aims to automatically analyze digital visual data. For instance, we want to recognize and localize different objects in an image. We further wish to describe their appearance and identify their geometric and semantic properties, as well as to characterize their role in the scene and relations among them. Solving these tasks would allow us to approach the long-standing objective of *holistic scene understanding*: processing the visual evidence at a level approaching the general understanding of humans.

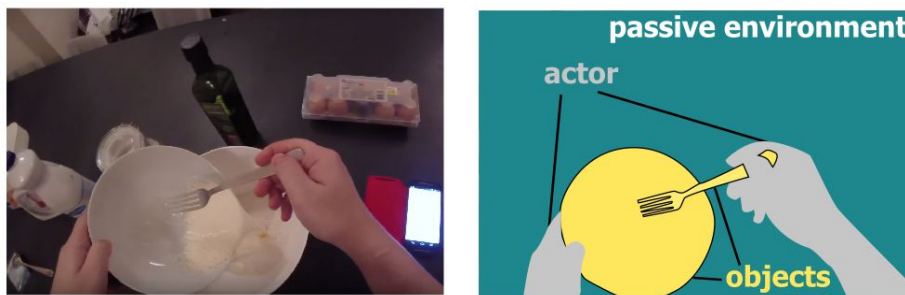


Figure 1-1: Scenes can typically be visually factorized at the pixel level into active agents, most often humans, objects and the surrounding static environment. We illustrate this factorization on an example image from the Epic-Kitchens 2018 dataset Damen et al. (2018).



Figure 1-2: Example of hand-object interactions which occur in a kitchen environment during food and drink preparation from the Epic-Kitchens 2018 dataset Damen et al. (2018). Object manipulation plays a crucial roles in activities of daily living (ADL).

Person analysis. Videos captured by and for humans grant them a large fraction of the visual space. An average of 35% of pixels in a set of consumer videos were attributed to humans according to Laptev (2013). Research on automatic video analysis has similarly been biased towards human-centric analysis. In this context, as we illustrate in Figure 1-1, scenes can most often be factorized into passive environment, active objects and actors. Efforts in computer vision research have typically focused on analyzing subsets of the scenes and their interactions. In particular, human-centric tasks such as action understanding and pose estimation have received dedicated attention. While full-body pose estimation has been the major focus, some specific body parts - the face and hands - have received increased and dedicated attention. This interest reflects the special role of these body parts, which are characterized by high density of tactile sensors and a crucial role in perceiving our environment and interacting with it Corniani and Saal (2020).

When we are about to perform an action, our gaze assesses the constraints imposed by the environment and anticipates object motions which will occur during manipulations Johansson et al. (2011). Consider the first-person perspective scenario, where a head-mounted camera captures person’s actions, see Figure 1-2. In such images we can easily recognize objects manipulated by human hands and understand the underlying intent: pouring the milk into the second bowl. The location and pose of hands and objects, as well as their appearance provide important cues about the ongoing action, and contribute to our understanding of the scene.

Importance of object manipulation. Most people develop increasingly complex and fine-grained control of their hands during childhood. From a primitive grasping reflex, through exploration and demonstration, infants learn to manipulate objects and use them as tools to achieve various tasks. Children learn how to use a spoon to eat typically in the second year of their life Connolly and Dalglish (1989) and master handling more complex objects such as scissors and shoe laces between the fourth and seventh year Dixon (2006). Following this progressive acquisition of fine motor skills, most adults use their hands seemingly effortlessly as they interact with their environment.

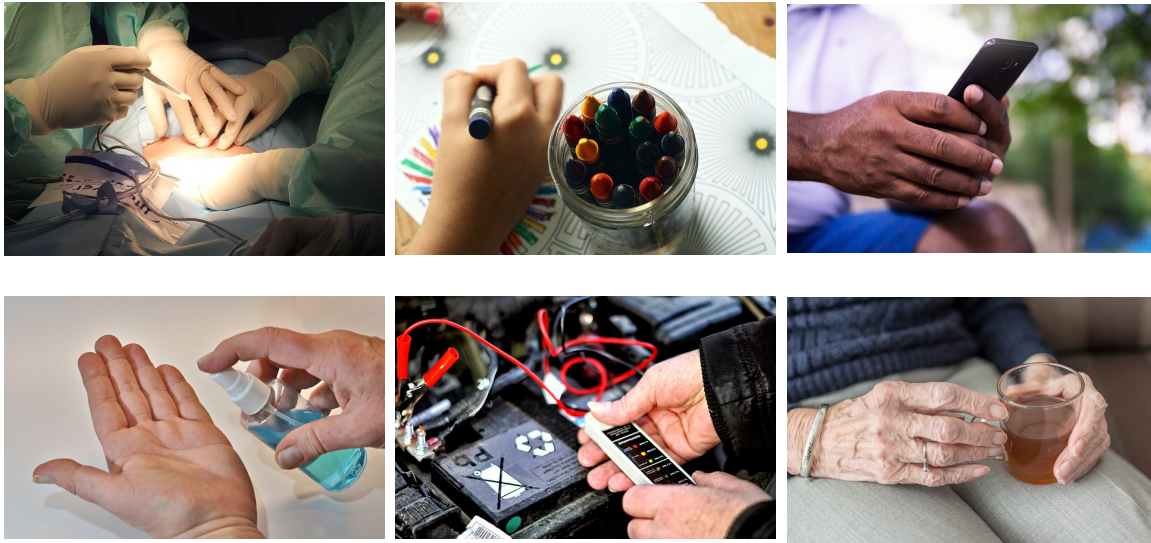


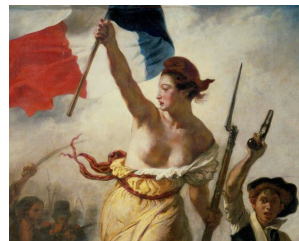
Figure 1-3: Various examples of object manipulation. People typically use their hands everyday to accomplish first-necessity tasks as well as for leisure and social activities. Images by Debora Alves, Sabine Ponce Joshua Woroniecki, Sabine van Erp, Renate Köpel, from pixabay.



(a) Detail of king Tutankhamun's canopic coffinettes, 14th century BCE, photo by Dmitry Denisenkov under the CC BY-SA 2.0 license



(b) Detail of Allegoria della Giustizia by Canova Antonio, 1792, photo from Fondazione Caripole Artgate under the CC BY-SA 3.0 license



(c) Detail of "Liberty guiding the people by" Eugène Delacroix, 1830, from Erich Lessing Culture and Fine Arts Archives via artsy.net



(d) Detail of "Liberty Enlightening the World", statue by Frédéric Auguste Bartholdi, 1886, photo by Petr Kratochvil

Figure 1-4: Famous examples of object manipulation.

Object manipulations represent a large fraction of most of our everyday activities. Humans, as well as some other primates, have the capacity to identify, use and store objects in order to perform dexterous object manipulation for everyday tasks as vital as eating and drinking Mulcahy and Call (2006). Compared with other species, humans use their hands to interact with a wider variety of objects that they create and shape to assist them in executing a wide diversity of tasks. We illustrate in Figure. 1-3 the wide range of tasks in which object manipulations play a central role. In addition to basic vital eating and drinking activities, people use objects to enhance their productivity, make use of passive or connected devices, and perform actions which require advanced control and dexterity such as surgery and art. In addition to their practical purpose, manipulations convey a symbolic meaning, as we illustrate in Figure 1-4. These observations motivate our work on automatic understanding of object manipulation.

1.1 Applications

In the following, we detail applications of hand-object modeling in Augmented Reality (AR) and in robotics .

1.1.1 Applications to AR.

Accurately estimating the pose of the hand and manipulated objects has applications in augmented reality, where computer-generated information enriches the user’s view of the world. The advent of consumer head-mounted displays (HMDs) such as the Google Glass Enterprise or Microsoft HoloLens which comes with a dedicated Computer Vision Application Programming Interface (API) (Ungureanu et al. (2020)) present opportunities for automatic training and assistance of users executing specialized tasks which involve object or device manipulations. The use of such wearable devices is especially relevant to train employees in tasks where their hands are involved, such as surgery Koo et al. (2021), as we illustrate in Figure 1-5a, or machine inspection and manipulation. Simulating critical situations creates an opportunity for education in a safe environment, and automated assistance during crucial tasks can result in higher quality interventions. Using HMDs during surgical operations can shorten interventions and thus reduce radiation exposure of the patient by overlaying guiding information in 3D in the physician’s field of view Jud et al. (2021). Accurately reconstructing the manipulation sequence through space and time could further allow to monitor the progression of interventions and to identify errors during execution.



(a) Use of augmented reality device during surgery. Image reproduced with permission from Khor et al. (2016).



(b) YuMi demonstrates pancake-cooking skills in a controlled environment.

Figure 1-5: Accurate modeling of unconstrained hand-object interactions could help provide guidance during manipulation using AR devices (see Section 1.1.1) or transfer useful skills to robots (see Section 1.1.2).

1.1.2 Applications to robotics.

Robots have become widely used in industrial controlled settings, where their adoption has resulted in increased productivity Nof (1999). Robots also demonstrate more versatility when the environment is sufficiently constrained, for instance when the shape of the manipulated objects is perfectly known as we illustrate in Figure 1-5b. In unconstrained environments such as our homes, fully automatic robotic assistance is mostly limited to a restricted set of tasks such as vacuum cleaning, mowing and fetching Zachiotis et al. (2018). Robots in this context are most frequently programmed to execute simple pre-defined steps, taking into account limited environment feedback to adapt to unseen configurations. Robots still lack the flexibility and robustness to be safely integrated in people’s homes, where they could assist people in activities of daily living (ADL). When asked, older people express the desire for physical assistance with demanding tasks such as picking up large and heavy objects as well as assistance for dexterous manipulations such as threading a needle or fastening jewelry pieces which are challenging for existing robotic systems Petrie and Darzentas (2017). Such tasks are neither easy to explicitly script or to formulate as control problems. Execution of such tasks, however, can be demonstrated by people. Learning from demonstration (LfD) Ravichandar et al. (2020) is therefore a compelling direction to explore to increase robotic versatility by implicitly learning the constraints and requirements of the task. LfD has been demonstrated in the context of manipulation using teleoperation Petrie and Darzentas (2011), for which an interface to control the robot in real time is required, or kinesthetic demonstrations: manually moving the robot parts Calinon et al. (2006). *Passive* demonstrations, where the user simply performs the intended task, can be easier to collect.



Figure 1-6: Everyday objects vary in shape. Even for a single seemingly constrained category such as mugs, a wide variety of shapes are observed in practice, see the first row. The variety of object shapes is reflected in CAD databases such as ShapeNet Chang et al. (2015) which source their models from the web, which we illustrate in the second row.

Reconstructing human hand-object interactions could thus be used to automatically guide robots in performing tasks in new environments and contribute to increasing the range of tasks they could tackle.

1.2 Challenges

Given an image of a scene, people can seamlessly identify individual objects, estimate object properties such as shape and weight, and to plan interactions. To achieve a similar level of capabilities, automatic systems need to face several challenges. In particular, one needs to design appropriate representations which can handle the diversity displayed by everyday objects (Section 1.2.1) and cluttered scenes (Section 1.2.2) where visual sensors provide only partial information (Section 1.2.3). Moreover, if adopting a machine learning approach, one needs to overcome a limited access to annotated data (Section 1.2.4).

1.2.1 Representing manipulated objects.

Everyday objects come with a high diversity of sizes and shapes (see Figure. 1-6). Hand-object interaction requires a contact between the hand and the object, and, hence, depends on the shape of the object surface. We are mainly interested in solid, potentially articulated or deformable objects common in everyday manipulations. Such objects have well-defined surfaces and an associated volumetric extent. Reasoning about the shape of 3D objects from sensor measurements requires the design of appropriate representations. We illustrate

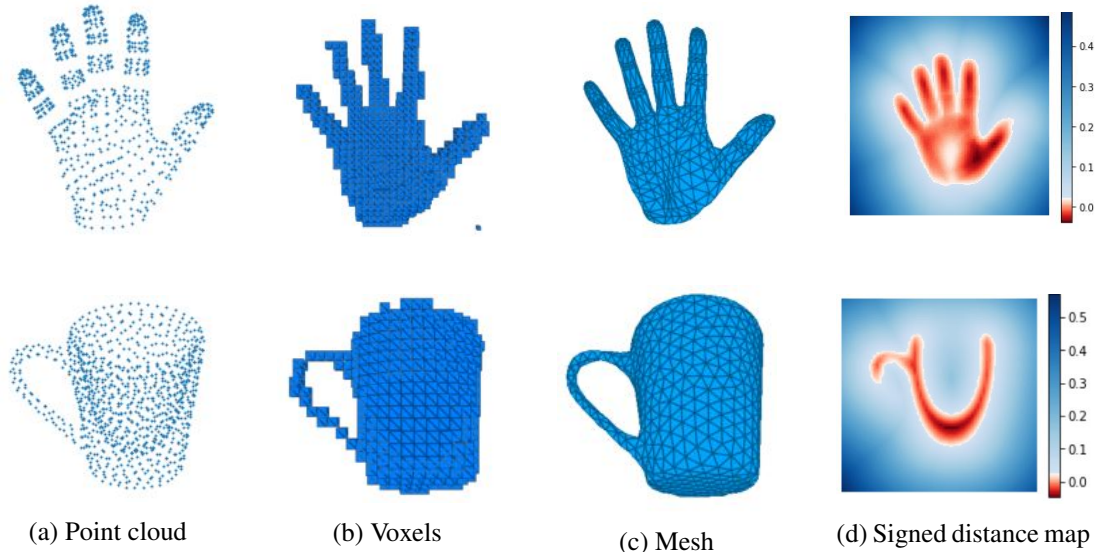


Figure 1-7: Different representations capture 3D information of material objects. 3D representations are discussed in more details in Section 2.1.1

several representations which have been explored in the context of data-driven object shape estimation in Figure 1-7. Different representations come with specific trade-offs in terms of precision, computational or memory requirements and relevance for a specific task. Modeling hand-object interactions requires choosing appropriate representations for both the hand and object. The chosen representations need to both allow for integration in optimization frameworks and be flexible enough to capture the diversity of possible objects. Furthermore, object representations need to be compatible with the hand representation to facilitate interaction reasoning.

1.2.2 Cluttered scenes

Hand-object interactions can be observed in many human-centric videos. Most of these videos are available in color (RGB) format. For instance, YouTube is the second most visited website and streamed more than 1 billion hours of videos to its users every day in 2020. 82% of YouTube users turned to video content on the platform for acquiring a new skill in 2020 according to Shalavi (2020). Be it learning to play chess or starting a garden, the two examples provided by the report on Youtube trends (Shalavi (2020)), these new skills involve manipulating objects such as physical chess pieces or a mobile device to play on virtual boards, and a variety of gardening tools. Such videos are challenging to process, as they typically present edited content with diverse viewpoints and scenes. When focusing on video datasets which specifically focus on capturing unconstrained object manipulations, such as the large-scale Epic-Kitchens Damen et al. (2018) dataset, videos often

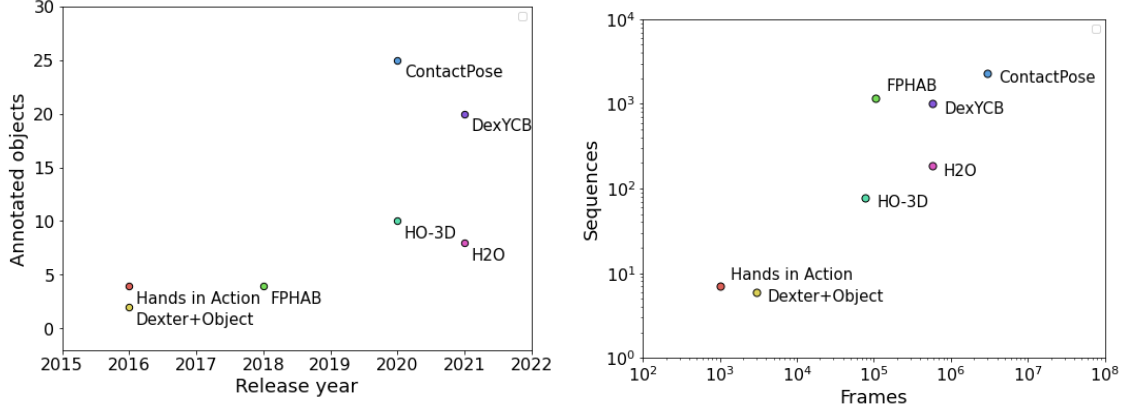


Figure 1-8: . Images and videos displaying natural hand-object interactions present many challenges. Note that difficulties often accumulate in a single image. The top right image for instance suffers from extreme illumination in addition to background clutter and occlusions.

present cluttered backgrounds, occlusions from the surrounding furnitures, extreme lighting conditions as well as motion blur. We illustrate these challenges in Figure. 1-8

1.2.3 Access to limited information.

A single-view video depicting hand-object interactions provides only partial evidence about the underlying manipulation. Crucial information relating to the object shape and hand pose is often unavailable due to self and mutual occlusions between the manipulating and manipulated entities in the scene. Given the articulated nature of the hand, the generated occlusion patterns can be complex. Working with RGB images makes the problem even harder, as pixel brightness is determined by the shape geometry, the unknown surface texture and lighting contributions. When the input is a single-view RGB image, the shape the object can not be exactly inferred. As occlusions are unavoidable in natural manipulations, part of both the object shape and hand pose are not visible, which can clearly be seen in Figure 1-8. Additionally the scale-depth ambiguity generates uncertainty in the localiza-



(a) Number of distinct objects for recently released hand-object datasets. (b) Number of distinct video sequences and frames.

Figure 1-9: Recent efforts have aimed to scale data annotation for real datasets depicting hand-object interactions. However, the total number of annotated objects and sequences is still limited.

tion of the action with respect to the camera. In this case, the most likely reconstructions must rely on strong priors for both the object shape and hand pose.

1.2.4 Limited 3D annotations.

Typical learning methods are supervised and require many input-output pairs for training. Learning-based approaches rely on labelled images to recover surface information from images. Furthermore, precise 3D annotations are crucial to evaluate and compare reconstruction methods. However, we currently still lack large-scale annotated hand-object interaction datasets. Manual annotation, which permitted impressive progress for tasks such as object recognition and segmentation following the release of the ImageNet Deng et al. (2009) and Pascal VOC datasets Everingham et al. (2015, 2010), is prohibitively tedious in the context of articulated bodies such as human hands. Over the past decade, efforts have been made to provide increasingly automatized annotation methods and to apply them to datasets of larger sizes. These methods currently still often rely on complex setups such as synchronized cameras capturing the scene from multiple angles which limit their deployment to a diverse scenes. While people touch on average more than a hundred objects on a single day, the largest annotated hand-object interaction datasets portray the manipulation of less than 30 distinct objects (see Figure 1-9). Additionally, the current capture methods which we detail in Section 2.3.3 generate biases in the captured data by restricting the range of allowed motions and/or the presence of visible sensors.

1.3 Goal

Our main objective is to automatically reconstruct the 3D geometry of hands and manipulated objects from color images. From a single RGB image or a short video clip depicting an object manipulation, we estimate dense surfaces of the object and the hand. We aim to accurately model interactions, capturing fine-grained finger poses and precise contact locations at the interface between the hand and object surfaces. To capture the versatility of human object manipulation, we focus on methods which can model interactions with a diverse set of objects.

Partial 3D information can be accessed directly using depth sensors, which record range data. However most images and videos which depict hand-object interactions are taken with standard color cameras. We want to learn from this large-scale data, hence, our methods need to handle color images. Recently, fully-supervised methods based on Convolutional Neural Networks (CNNs) have become the dominant approach for 3D reconstruction from RGB inputs. Methods which produce accurate estimates in real or near-real time during inference have been proposed for the related tasks of estimating the shape and pose of humans (Kanazawa et al. (2018a); Pavlakos et al. (2018)) and objects (Groueix et al. (2018b); Kehl et al. (2017); Labbé et al. (2020); Li et al. (2018); Rad and Lepetit (2017); Wang et al. (2018); Xiang et al. (2018)). Inspired by these successes, we investigate learning-based methods for hand-object reconstruction. While data-driven approaches typically require a large number of annotated samples, acquiring images with 3D information describing object manipulation is a challenging task, as we will discuss further in Section 1.2.

Given the annotation difficulties, annotated datasets are limited in size and diversity. This observation guides our focus towards methods which explicitly target data scarcity. We propose both learning and fitting-based methods for direct reconstruction of hands and objects, and investigate how to leverage alternative sources of annotations which are easier to collect at scale. More specifically, we propose to investigate the use of synthetic data in Chapter 3, temporal context in Chapter 4 and separate hand and object noisy annotations in Chapter 5.

1.4 Contributions

In this thesis, we propose to model hand-object interactions from color images in both learning and fitting frameworks. Given the 3D label scarcity, we explore different sources of supervision which are more readily accessible. Manipulations impose anatomical and interaction constraints during manipulation. We take advantage of this fact and investigate

how such constraints can be leveraged to recover more plausible reconstructions of hands and manipulated objects.

In Chapter 3, we present the first end-to-end differentiable method for joint hand-object reconstruction from a single RGB frame. We integrate a parametric hand model as a differentiable layer in our neural network and estimate the object shape by learning to deform a spherical mesh template. We regularize the reconstruction of hands and objects with manipulation constraints. Our learnable model exploits a novel contact loss that favors physically plausible hand-object interactions. We show that our approach improves grasp quality metrics over baselines. To train and evaluate the model, we also propose a new large-scale synthetic dataset, ObMan. We demonstrate the transferability of models trained on ObMan to real data.

In Chapter 4, we present a method to leverage photometric consistency across time when annotations are available only for a sparse subset of frames in a video. Our model is trained end-to-end on color images to jointly reconstruct hands and objects in 3D by inferring their poses. Given our estimated reconstructions, we render the optical flow between pairs of images and use it to warp one frame to another. Both the rendering and warping are differentiable operations, allowing the supervision to be added as an additional loss term during training. Our self-supervised photometric loss relies on the visual consistency between nearby images. We demonstrate that our approach allows us to improve the pose estimation accuracy by leveraging information from neighboring frames in low-data regimes. This work achieved state-of-the-art results at the time of publication on 3D hand-object reconstruction benchmarks.

In Chapter 5 we propose to leverage predictions from trained models to guide joint hand-object fitting. We go beyond single hand pose estimation, allowing us to model more complex 2-hand manipulations, and reconstruct the action over a short video clip. We rely on recent learnt models for hand-object detection, 3D hand pose estimation and object segmentation to provide constraints within an optimization framework. In addition to visual evidence, we integrate smoothness and interaction priors to direct our optimization towards plausible hand-object reconstructions. We show that our method consistently recovers plausible interactions in favorable viewing conditions and present promising results on more challenging datasets.

1.4.1 Publications

The work during this PhD led to the following publications:

- Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev and Cordelia Schmid. Learning joint reconstruction of hands and manipulated

objects. In CVPR 2019. Hasson et al. (2019)

- Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys and Cordelia Schmid. Leveraging Photometric Consistency over Time for Sparsely Supervised Hand-Object Reconstruction. In CVPR 2020. Hasson et al. (2020)
- Yana Hasson, Gül Varol, Ivan Laptev and Cordelia Schmid. Towards reconstructing unconstrained hand-object interactions. *Under review*, 2021. Hasson et al. (2021)
TODO: Add link

1.4.2 Software & dataset contributions

The work done during this PhD resulted in the following code releases:

- ObMan: Scripts to generate our synthetic dataset ObMan which implements the rendering approach detailed in Section. 3.2. https://github.com/hassony2/obman_render
- Code and trained models for hand-object reconstructions for the work presented in Chapter. 3. https://github.com/hassony2/obman_train.
- Code and trained models for our sparsely-supervised hand-object reconstruction method presented in Chapter. 4. <https://github.com/hassony2/handobjectconsist>.
- Fitting scripts for estimating the poses of the hands and manipulated object for a short RGB video clip, see Chapter. 5. **TODO: ADD code link**
- A port of the MANO Romero et al. (2017) hand model to PyTorch Paszke et al. (2019). <https://github.com/hassony2/manopth>
- Conversion scripts which transfer the weights of action recognition models from the TensorFlow (Abadi et al. (2015)) implementation of I3D convolutional models from the paper Carreira and Zisserman (2017) to PyTorch (Paszke et al. (2019)). https://github.com/hassony2/kinetics_i3d_pytorch.
- Scripts to adapt generic ResNet He et al. (2015) and DenseNet Huang et al. (2017) 2D CNNs to take video as inputs in the context of video classification. https://github.com/hassony2/inflated_convnets_pytorch applying the method described in Carreira and Zisserman (2017).
- A toolbox for video data-augmentation in Python. https://github.com/hassony2/torch_videovision.

ObMan dataset. We have publicly released the ObMan dataset (<https://www.di.ens.fr/willow/research/obman/data/>) presented in our publication Hasson et al. (2019) in collaboration with the Max Planck Institute. ObMan consists of synthetic images with automatically generated object grasps. The generated images come with rich annotations including depth data, segmentation masks for the hand and the object and the underlying 3D meshes for the human and grasped object.

1.5 Outline

This thesis consists of 6 chapters including this introduction. Chapter 2 reviews the literature related to the task of hand and object modeling from visual data. Chapters 3, 4 and 5 present our contributions in hand-object modeling from RGB images. In Chapter 3 we explore learning from synthetic data and enforce physical interaction constraints at train time. We then investigate consistency constraints over neighboring frame as a weak form of supervision in Chapter 4. In Chapter 5.2.3 we focus on fitting known hand and object models to automatically collected noisy labels. We conclude in Chapter 6.2 with a summary of contributions, open problems and a discussion of promising future work directions.

Chapter 2

Related work

This chapter provides a survey of previous work in object pose and shape estimation. We first focus on independent object modeling in Section 2.1. We then focus the human hand, and review methods which reconstruct hand shapes and poses in Section 2.2. Finally, we review methods which model hands and objects simultaneously in Section 2.3.

2.1 Object modeling

Object

1. Something material that may be perceived by the senses.

Merriam-Webster dictionary (2021)

Objet

1. Tout ce qui affecte les sens et, en particulier, tout ce qui s'offre à la vue, au toucher.
2. Chose, réalité matérielle, destinée à un usage précis.

Dictionnaire de l'Académie Française
(2021)

In this section, we first introduce representations which have been used to describe 3D objects in the context of visual modeling (2.1.1). We then review works which estimate the shape of objects from visual evidence (2.1.2) and finally highlight methods which estimate

the object rigid pose when the object shape is known (2.1.3).

2.1.1 Object representation

As appears from the definition, material objects are of specific interest because we can sense and interact with them. A useful notion to formalize object interactions is the notion of object boundary or surface. While at the sub-atomic level object boundaries are ill-defined Varzi (1997), they practically characterize the spatial extent of the object at the macroscopic scale. We will work with the view that solid objects have a volumetric extent and that their surfaces can in general be represented by continuous closed 2D manifolds embedded in 3D without self-intersections.

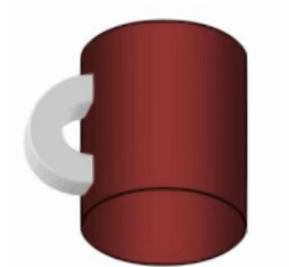
We highlighted in Section 1.2.1 that everyday objects come in arbitrary shapes. This implies that an infinite number of parameters is theoretically needed to express the full range of possible 3D shapes Anguelov (2005). In effect, everyday objects which are created naturally, accidentally, or intentionally, display regularities which restrict their diversity.

A good object representation should take into account the constraints and objectives of the target task. Several useful representations for 3D objects have been developed and used in the context of graphics, design, robotics and computer vision. When targeting application scenarios such as efficient rendering, 3D printing, grasp planning or reconstruction, task-specific constraints are imposed on the object’s representation. Real-time processing or the need to encode local information such as surface normal directions or part labels might be practical requirements. In the context of hand-object modeling, we need to choose an object representations which encodes critical spatial features while also affording efficient collision checking to reason about physical constraints and plausibility. Reconstructing the interaction in real time increases the range of possible downstream applications.

In the following, we will provide an overview of different representations (primitives, meshes, point clouds voxels and Signed Distance Maps (SDMs)) and specifically investigate their amenability to capture objects which we routinely interact with in our daily lives. In particular, we will consider the trade-offs imposed by each representation. Choosing a given representation constrains the flexibility, for instance the possibility to model topological variations and the level of surface detail. A specific representation is also associated with specific storage as well as computational costs for basic operations. In the context of interaction modeling, we will examine the computational burden associated with collision checking. In particular, determining whether a point is on the inside or outside of the modeled object, and retrieving the distance of a point to the object’s surface.



(a) Image (synthetic rendering)



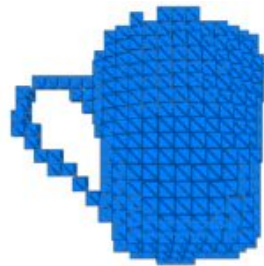
(b) Primitive decomposition Biederman (1987)



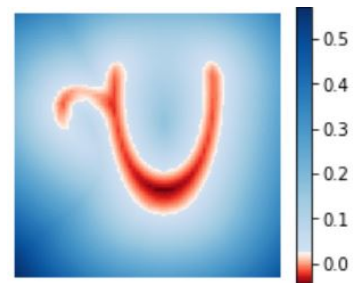
(c) Mesh



(d) Point cloud



(e) Voxels



(f) SDM

Distances to the object surface are color-coded in red inside and in blue outside the object.

Figure 2-1: Different 3D representations allow to capture and encode a given 3D shape, such as this mug from the ShapeNet Chang et al. (2015) dataset. Each representation incurs a different trade-off in terms of flexibility, modeling capacity and storage costs, determining its adequacy for a given task. 3D representations are discussed in more details in Section 2.1.1

Primitives. Simple shapes which have compact representations can be combined to approximate the shape of an object. Several primitives have been used in this context such as spheres, which are characterized by their center and radius, cubes or pyramids. More complex 3D shapes such as superquadrics Barr (1981), a family of 3D shapes whose surfaces are defined by implicit equations on 3D coordinates, offer more flexibility and allow to approximate the surfaces with greater precision. Primitive approximations of 3D shapes have practical advantages. In most cases, primitive decompositions can be efficiently stored and manipulated. For instance, primitives which can be expressed using implicit equations on point coordinates typically allow collision checking in constant time for arbitrary points. For simple shapes such as spheres and cubes, computing distances to the surface for 3D points can also be achieved in constant time. Such efficiency often comes at the cost of limited accuracy and results in coarse surface modeling which we illustrate in Figure 2-1b: using a restricted number of instances from a limited set of candidate primitives allows to roughly approximate a the mug’s shape. This trade-off can be partially addressed by using a larger family or by increasing the number of primitives used in the decomposition.

Meshes. In the context of graphics and robotics, object shapes are often stored as polygonal meshes, typically defined by a set of *vertices*, 3D points on the object surface, and *faces*, an ordered set of vertices, which define convex polygons on the object’s surface. Meshes (Figure. 2-1c) are typically decomposed into triangular surface patches, each face delimited and defined by exactly 3 vertices. Given enough vertices and faces, a mesh can approximate a physical object’s surface with arbitrary precision. An advantage of meshes for capturing the object shape is their potential compactness. Vertex density can be adapted across the object’s spatial extent to account for different level of details. Large flat or approximately flat regions can be represented using a sparse set of points while regions with intricate patterns can be precisely modeled using an appropriately increased density of vertices. A potential downfall of meshes is that they do not directly capture the volumetric extent, which can be useful for downstream tasks such as checking collision between two meshes. Computing the distance from a point to a mesh typically requires computing the distance to all the faces Akenine-Möller and Trumbore (1997). Additionally, meshes are constrained by their underlying triangulation. Any fixed template cannot be deformed into a target object as a fixed triangulation prevents topological variations. Triangle meshes are flexible, and can encode shapes beyond the scope of physically realistic objects, including self-intersecting 3D shapes. A subset of meshes of specific interest to us is the set of watertight meshes, for which the interior and exterior are well-defined, and can be considered as containing the set of physical objects.

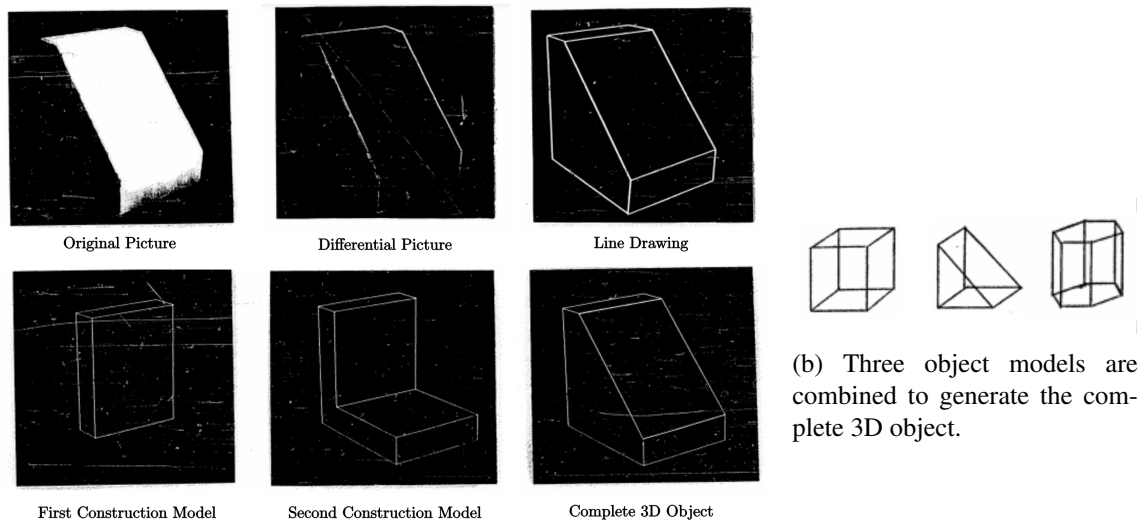
Point clouds. An alternative restricted representation of objects is point clouds (Figure. 2-1d), a collection of point coordinates sampled on the object surface. Point clouds provide indications on the spatial arrangement of the object. Similarly to meshes, the number of points can be increased to capture precise details or reduced to store a lightweight set of keypoints. However, point clouds do not explicitly encode the full 3D shape. The surface and volumetric extent can only be partially recovered from this representation, for instance using local heuristics for normal computation and Screened Poisson reconstruction Kazhdan and Hoppe (2013). As point clouds can be obtained directly from measuring devices, no conversion is needed and the information can be preserved and used directly to describe the target objects.

Voxels. Object shapes can also be encoded as voxels, a representation of the object's volumetric occupancy on a discrete grid in 3D space. Voxels are typically stored as binary values for a uniform discrete grid. Instead of storing the surface information, voxels (illustrated in Figure. 2-1e) directly encode the presence or absence of the object. This representation can be convenient for applications which require reasoning about the object's volumetric extent. Voxels typically involve important memory costs, as the number of cells increases cubically as a function of the grid resolution. This limitation can be mitigated using efficient implementations such as sparse octrees Laine and Karras (2011). Volumetric discretization prevents direct access to the object's surface, which can however be approximately recovered, for instance using the Marching Cubes algorithm Lorensen and Cline (1987) which converts a voxel representation to a mesh.

Signed distance maps (SDMs). Signed distances provide an alternative encoding of object shapes Malladi et al. (1993). SDMs associate to any 3D location the distance to the object's surface, oppositely signed by convention on the outside and inside of the object. Signed distances can be stored on a voxel grid, in which case they allow to capture more details compared to a voxel grid of the same resolution, or can be expressed as Signed Distance Functions (SDFs) which are continuous functions parametrized by the 3D space coordinates. Similarly to voxels, this representation doesn't provide explicit access to the object surface, which can however be recovered with appropriate processing. Figure 2-1f displays a visual representation of the SDM of a mug computed on a square surface slice.

2.1.2 Object shape estimation

Object shape estimation is a long-standing tasks in computer vision and has a rich history. In the following, we will review work that focus on recovering the geometry of objects from color or gray-scale images and specifically emphasize recent data-driven approaches.



(a) Roberts (1963) processed grayscale images to extract the lines which match projected edges and combined primitive shapes to reconstruct the target object shape.

Figure 2-2: One of the first attempts to automatically recover the shape of 3D objects from 2D images by Roberts (1963). Images from Roberts (1963)

We will first review methods which recover the object geometry by decomposing them into primitive shapes (2.1.2). Then we will present work that propose to estimate the volumetric extent of the object using voxels or SDMs (2.1.2), and efforts to retrieve object models from databases given visual evidence (2.1.2). Finally, we will present methods which approximate objects by deforming mesh templates (2.1.2).

Approximating object shapes using primitive shapes. The task of estimating the 3D shape of objects from two-dimensional photographs using a computer can be traced back to Lawrence Gilman Roberts's thesis Roberts (1963). This approach marks the beginning of a line of work which focuses on recovering objects as a collection of 3D primitives from 2D images. In this seminal work, as illustrated in Figure 2-2, an image is first processed to recover lines which correspond to edges of a 3D object. Following the heuristic that an object can be seen either as an instance of a known model or can be decomposed into known parts, three basic models are assembled to generate a composite shape which explains the observed contours. After this initial attempt, different approaches model the world using sets of planar surfaces Guzman (1968); Kanade (1981) or simple primitives such as cylinders Marr and Nishihara (1978)

Several early work perform shape estimation using more complex primitives. Binford (1971) introduce generalized cylinders, which are defined by a curved axis and a sweeping rule which describes how the cross-section evolves along the curve. Superquadrics Barr

(1981), have alternatively been used in the context of object shape estimation by Pentland (1986). Biederman (1987) proposes to decompose objects into geons, a fixed dictionary of object shapes which often occur in human-made objects. A mug, for instance, can be reasonably decomposed in only two geons, a cylinder and a handle as illustrated in 2-1b. Most of these early methods rely on computing edges and explicitly reasoning about the constraints they impose on a target 3D shape Mundy (2006).

Recently, learning-based methods have been the main paradigm for single view shape reconstruction (SVR). Observing that many objects (among which printers, books and furniture) can be roughly approximated using cuboids, Fidler et al. (2012) and Xiao et al. (2012) propose to recognize and estimate the pose of rectangular-shaped objects in a scene. Del Pero et al. (2013) generate more complex non-convex object models by combining simple primitives and reconstruct scenes containing objects such as chairs and tables. Xiang and Savarese (2012) model objects by composing planar elements and adapt a category-specific template to visual evidence, effectively retrieving a coarse approximation for the main structural elements of the observed object. Their optimization procedure allows the planar elements' location and shape to vary from the initial template, enabling instance-specific shape fitting. These methods allow to model a wider set of objects but require manually or semi-automatically constructing category-specific templates, restricting in practice the number of modeled categories to 12 for Xiang and Savarese (2012) and 8 for Del Pero et al. (2013).

Recent methods removed the requirement on manually defined object templates by training neural networks to directly predict primitive parameters from color images Niu et al. (2018); Paschalidou et al. (2019); Tulsiani et al. (2016) as we illustrate in Figure 2-3. Tulsiani et al. (2016) propose to reconstruct the shape of objects by learning to predict a variable number of cuboid locations, orientations and extent given an input image. They use a CNN to directly estimate a variable number of primitives and provide a coarse approximation of the observed object. Niu et al. (2018) also model the object shape from RGB images using cuboid primitives, additionally modeling structural information such as part connectivity. Paschalidou et al. (2019, 2020) show that the surface of objects can be modeled with improved accuracy by predicting the parameters of superquadrics Barr (1981). Using an increased number and more diverse primitives allows for better approximations of the object surface. Current learning-based methods rely on 3D annotations which are available only for a subset of everyday objects in practice.

Occupancy estimation. Modelling objects from a single view is an ill-posed problem and requires a prior on the geometry of occluded regions. Such a prior can be learnt from data in practice, a line of work which has been accelerated by the recent success of neural

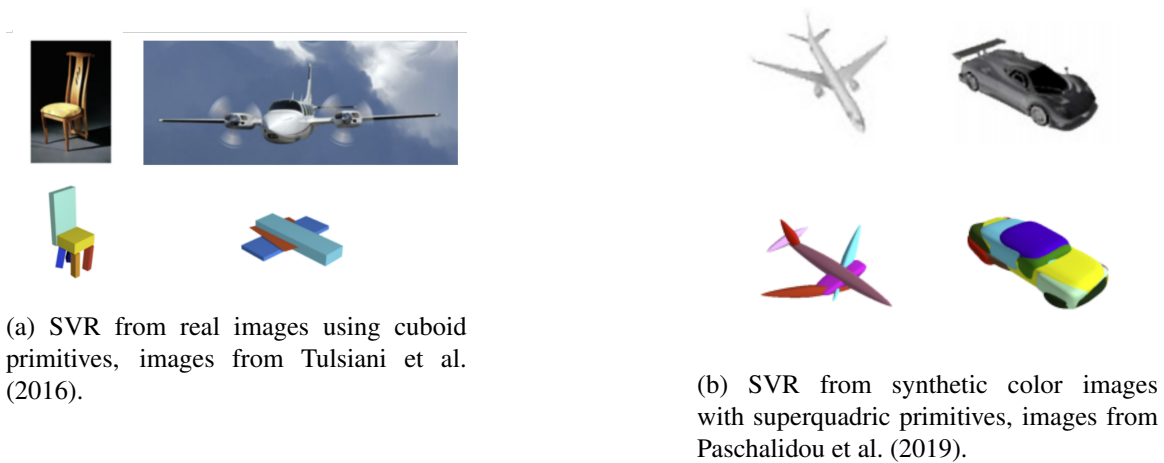


Figure 2-3: Recent learning-based methods demonstrate SVR results for real and synthetic images by recovering the parameters for a variable number of shape primitives.

networks. One of the first learnt methods which estimates the complete volumetric extent of an object from a single image, Choy et al. (2016), predicts voxel occupancy from image features using 3D convolutions. Predicting voxel occupancy allows to approximate objects of arbitrary topology, and can be applied to all existing objects. However, regressing occupancy for each voxel location scales cubically with the desired spatial resolution, making direct regression impractical for fine details modelling. Several following work reduce the memory requirements for this representation, for instance Riegler et al. (2017) use sparse octrees, an efficient implementations of voxels Laine and Karras (2011), and leverage the fact that occupancy only needs to be refined if a voxel at a coarser resolution is occupied. A drastically different approach mitigates the memory requirement by regressing voxel occupancy as a function of the coordinate’s location Mescheder et al. (2019), removing the dependency on expensive 3D convolutions. Similarly, several concurrent work use neural networks to regress the signed distance to the object surface for any arbitrary point location Chen and Zhang (2019); Park et al. (2019a). Improved qualitative results are demonstrated by using both global and local image features when approximating the SDF as a neural network Xu et al. (2019). Methods which encode objects as SDFs have the advantage of providing direct collision checking, recovering arbitray shape topologies as well as fine details, as illustrated in Figure 2.1. However, computing signed distances requires expensive forward passes through the learnt neural networks, and a large number of such evaluations can be needed in practice to recover a good approximation of the object’s surface.

Shape estimation by model query. The emergence and standardization of Computer Assisted Design (CAD) has led to an increased availability of CAD models which capture the geometry of diverse objects Corney et al. (2005). The availability of such models motivated a line of research which frames object reconstruction as object model query from a CAD model database. Malisiewicz et al. (2011) obtain posed object models as a byproduct of their exemplar matching approach. They train a per-training-sample SVM classifier for object recognition and transfer a posed 3D CAD model associated to the predicted image to the test sample. Aubry et al. (2014) estimate the object shape by matching real images with synthetic renderings of models from curated databases. Mottaghi et al. (2015) merge several CAD models for a given subcategory such as race-cars into a template, and combine sub-category prediction with continuous pose estimation, recovering a posed object model from a single RGB image. Starting from a monocular RGB image, Bansal et al. (2016) predict image normals and use this predicted output along with the color image to align CAD models to the visual evidence.

The release of large-scale CAD datasets such as ShapeNet Chang et al. (2015) and ModelNet Wu et al. (2015), provided a standardized test-bed for single and multiple-view shape reconstruction tasks. More recently, joint model query and pose estimation have been revisited using Deep Neural Networks (DNNs) by matching synthetic renderings and real images using deep learnt features. Wohlhart and Lepetit (2015) use robust learnt features to outperform hand-crafted descriptors for the task of object recognition and 3D pose estimation. They formulate the prediction of the class and pose of an object in an image as a nearest-neighbor query for a set of object instances from the LineMOD dataset introduced by Hinterstoisser et al. (2012b). Starting from an RGB image, Grabner et al. (2018) perform query in the larger ShapeNet Chang et al. (2015) database, recovering posed model candidates for unseen instances for target object categories for which training data is available. They show improved performance by matching using location fields features, an intermediate representation which associate pixel locations to 3D surface coordinates of objects in canonical pose Grabner et al. (2019). Tatarchenko et al. (2019) propose a retrieval baseline for single-view shape reconstruction. After embedding 3D shapes in a 512-dimensional space, they learn to predict shape descriptors from image inputs using a CNN.

Shape from model deformations. Querying from a fixed database limits the modelling accuracy by restricting the reconstructions to a pre-determined number of candidates. In contrast, an infinite diversity of object shapes can be recovered by continuously deforming point clouds or mesh templates. Several work focus on modelling variations between different instances of specific object categories such as cars and bicycles Kar et al. (2015); Zia et al. (2014), or fish Prasad et al. (2010). Kar et al. (2015) learn category-specific point

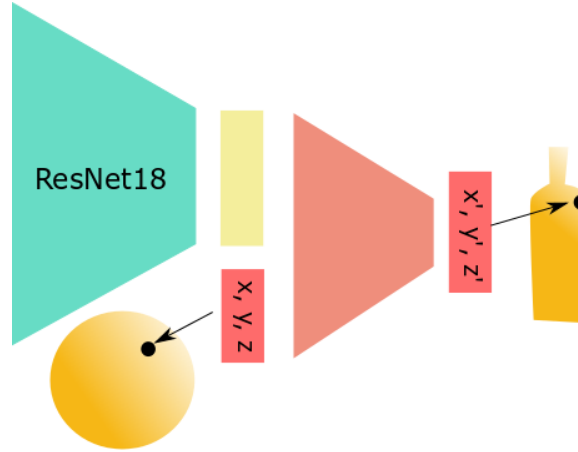


Figure 2-4: AtlasNet Groueix et al. (2018b) learns to deform a template by mapping its surface points to surface point coordinates of the reconstructed object.

cloud deformations to model intra-class variability, demonstrating that they can recover 3D shape information using paired images and 2D masks for supervision. Such methods bypass the need for pairs of associated images and 3D models and instead rely on the availability of precise 2D information such as segmentation masks in the case of Kar et al. (2015); Zia et al. (2014) or contours for Prasad et al. (2010).

Wang et al. (2018) propose to deform a spherical mesh using a deep neural network (DNN), by propose to use Graph Convolutional Networks (GCNs) for this task. They demonstrate that their method can reconstruct fine details. However, relying on a simple spherical template reduces the range of possible objects which they can model. Given image features, Groueix et al. (2018b) learn to deform simple templates - a sphere or a set of square surface patches - using a multi-layer perceptron (MLP). For each point on the template, represented by its 3D coordinates, the network learns to predict a new 3D position on the surface of the observed object, as illustrated in Figure 2-4. Using more flexible templates such as sets of square patches allows them to reconstruct objects of complex topology, for instance chairs which often have several holes in the back, but results in non-watertight meshes. Groueix et al. (2018b), Deprelle et al. (2019) and Wang et al. (2018) use ground truth 3D supervision to train their SVR models. Such annotations are typically only available at scale for synthetic datasets.

Representing the object as a mesh allows to explicitly model visual appearance at the pixel level through a rendering step. Rendering automatically generates 2D images for scenes with given shapes, textures, shading and lighting models. Typically, rendering is a non-differentiable operation because of the rasterization step, which assigns a unique observed surface to each pixel location. Several differentiable approximations of the rasterization process have been proposed to allow for differentiable rendering, which allows

to compute the derivatives of pixel color or segmentation values with respect to the scene inputs (Chen et al. (2019); Kato et al. (2018); Liu et al. (2019); Loper and Black (2014)). Liu et al. (2019) propose to approximate the rendering step through *soft* rasterization. Unlike traditional rasterization, the final pixel color combines contributions of different faces which project close to a target pixel location. The final color depends of the face’s distance to the camera and of their distance to the pixel location. In contrast, OpenDR (Loper and Black (2014)) and Neural Mesh Renderer (NMR, Kato et al. (2018)) propose to perform exact rasterization in the forward pass and propose smooth approximation to the discrete rasterization at the triangle face boundaries in the backward pass. While Liu et al. (2019) potentially distributes pixel gradients across several faces, NMR and OpenDR gradients flow back only to the the face rendered in the forward pass. All the described differentiable renderers allow to directly tune the object parameters, for instance vertex locations, to account for the observed segmentation masks or pixel values. This optimization can be performed for a single image or segmentation mask or use multi-view information. A single view necessarily provides incomplete cues for the final object shape when the dimension of the optimized parameter space is large (for instance when optimizing each vertex location independently). Additional mesh regularization can somewhat improve the shape estimate, as we illustrate in Figure 2-5c. Typically, Laplacian regularization Nealen et al. (2006); Pinkall and Polthier (1993) or face normal smoothness regularization can limit strong variations in the mesh curvatures while edge length regularization favors more uniform triangulations. However, stronger priors which take into account semantic information are needed to recover plausible object shapes, see Figure 2-5

Differentiable rendering has been integrated in learning frameworks in order to leverage the supervision from segmentation masks. Kanazawa et al. (2018b) reconstruct a specific category: birds, using differentiable rendering. They train a model to provide both an estimate of the camera viewpoint, the bird’s shape and the texture using manually annotated keypoint as well as mask and texture losses based on the outputs of NMR Kato et al. (2018) for supervision. While using only keypoints and color image supervision during training, they can estimate the 3D shape as well as texture of the target animal from a single RGB image at inference time. Later work removes the dependency on keypoints Goel et al. (2020) by maintaining a set of possible camera hypotheses instead of a single viewpoint prediction. These methods allow to learn template deformations without relying on costly 3D annotations, but learn category-specific priors, and are restricted to modeling shapes of topology constrained by a template.

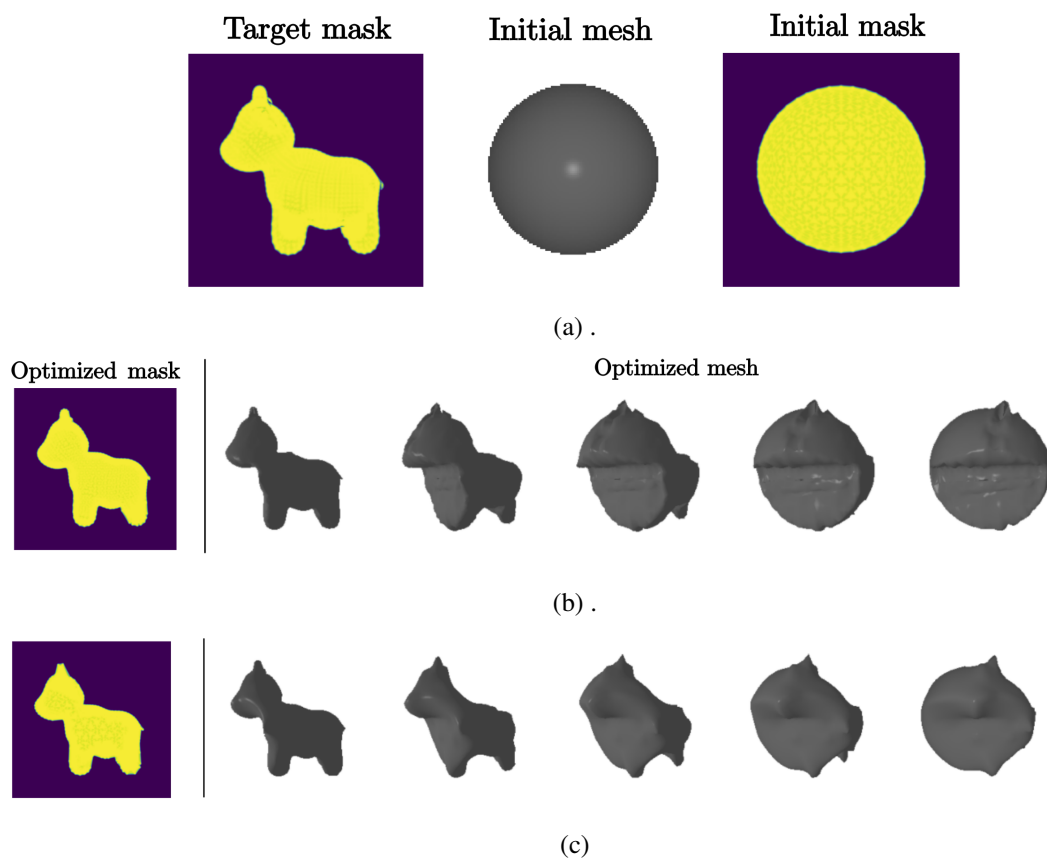


Figure 2-5: Differentiable rendering allows to deform a mesh template so that it matches a given silhouette (2-5a). Here we display optimization results where we fit a spherical mesh to a target shape silhouette by optimizing each vertex location independently using Pytorch3D. While the outline imposes valid constraints on the object shape, the problem is underconstrained in the case of single-view silhouette optimization, resulting in implausible shapes (2-5b). Additional regularization constraints, in this case a weighted sum of Laplacian, normal smoothness and edge length regularization (with weights 1, 0.01 and 1 respectively, as per the Pytorch3D Ravi et al. (2020) tutorial) provide an improvement but fail to capture semantic priors (2-5c).




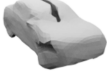














Input Image	GT	3D-R2N2 Choy et al. (2016)	Pix2Mesh Wang et al. (2018)	AtlasNet Groueix et al. (2018b)	OccNet Mescheder et al. (2019)
					
					
					

Table 2.1: SVR results for recent CNN-based methods. 3D-R2N2 Choy et al. (2016) can reconstruct arbitrary objects but with a limited resolution. Pixel2Mesh Wang et al. (2018) produces detailed results but is restricted to a simple topology as it deforms a sphere template. AtlasNet Groueix et al. (2018b) reconstructs complex topologies but results in non-watertight meshes. Occupancy prediction Mescheder et al. (2019) results in precise watertight meshes but requires post-processing to extract the surface. Images courtesy of Mescheder et al. (2019).

2.1.3 Rigid pose estimation

When the object model for the observed rigid object is known, the rotation and translation of the model fully capture the state of the object in the scene. Rigid pose estimation estimates the object pose given a fixed camera (or symmetrically, the camera viewpoint for a fixed object) for a known object model. Several approaches have emerged and have been revisited over the years to solve this task. A dominant approach relies on predicting 3D information in the 2D pixel plane and solving for the optimal pose in a subsequent step (2.1.3). Alternatively, template matching (2.1.3) tries to recover the object pose by matching a representation of the image with a representation of the posed 3D object model. Leveraging the versatility of neural networks, recent methods propose to directly regress the object rotation and translation from the image pixels (2.1.3). Recently, further improvements have been demonstrated using refinement strategies from coarse pose estimates (2.1.3).

Feature-based 2D-3D correspondences. Similarly to early work in object shape estimation, early efforts in pose estimation attempt to align projected edges of known 3D models to contours extracted from images Mundy (2006). Provided a CAD model is available, Huttenlocher and Ullman (2020) aligns the object model of a flat object to its photograph by observing that 3 pairs of matching points are sufficient to recover the object’s scale, orientation and position under the orthographic projection model. Lowe (1987) propose

to detect and group contours to estimate the viewpoint of object models. Going beyond rigid pose estimation, the described method can be extended to estimate unknown model parameters such as angles or part lengths. Dhome et al. (1989) matches a triplet of image lines between the object model and its projection to recover the 3D pose of a known model.

Several work propose to rely on *local* features and recover 2D-3D correspondences before subsequently recovering the 3D object pose using a variant of a RANSAC-based Fischler and Bolles (1981) Perspective-n-Point (PnP) algorithm Moreno-Noguer et al. (2007). Rublee et al. (2011) for instance introduce ORB, hand-crafted local features which can be computed efficiently. They evaluate their local descriptors on the task of object detection and rigid pose estimation using EPnP Lepetit et al. (2009), providing comparisons with other mainstream local descriptors such as SIFT Lowe (2004) and SURF Bay et al. (2006). Relying on local features however limits the performance of the proposed methods when applied to texture-less objects.

Given 2D detection annotations, CNNs can successfully learn to detect 2D keypoints for texture-rich as well as texture-less objects by processing both local and global appearance cues. Several methods propose to use a CNN to detect 3D object corners in 2D images. Rad and Lepetit (2017) propose BB8, a two-stage approach which first segments objects and uses the segmented output to predict projected locations of 3D bounding boxes as a second step. While effective, their method does not run in real time because of its multi-stage nature. Oberweger et al. (2018) showcase that predicting keypoint heatmaps is sensitive to occlusions, and propose to aggregate predictions across several image patches for improved robustness. This performance improvement comes however at the expense of an increased runtime. Several subsequent work attempt to predict the projected bounding box locations in a single CNN forward pass by adapting architectures developed for single-shot object detection and segmentation such as SSD Liu et al. (2016) or YOLO Redmon et al. (2016). Tekin et al. (2018) propose a method based on YOLO Redmon et al. (2016) which regresses project bounding boxes coordinates for each detection computed on a discretized pixel-aligned grid. They show competitive 3D pose results using PnP. By avoiding the computational burden required by refinement steps, they reach real-time performances at test time.

Park et al. (2019b); Peng et al. (2019) propose to predict 3D information at each 2D object location. Alternatively to 3D bounding box location regression, Brachmann et al. (2016) regress 3D object coordinates at pixel locations, a parametrization introduced by Brachmann et al. (2014), from RGB inputs using random forests Breiman (2001) with auto-context Tu and Bai (2010). They obtain a pose estimate using RANSAC Fischler and Bolles (1981) and further refine it using the predicted outputs. Peng et al. (2019) introduce a Pixel-wise Voting Network (PVNet) to densely regress sparse keypoint positions a

each object pixel location. Instead of predicting bounding box locations, which they observe lead to high variance in the predictions, they automatically define keypoints on the object surface for each object. They initialize the final object pose using EPnP Lepetit et al. (2009) and further refine it by taking into account the densely predicted spatial probabilities. Park et al. (2019b) propose Pixe2Pose which regresses 3D coordinates in the pixel space as in Brachmann et al. (2014, 2016) starting from an image and 2D detections. They regress 3D coordinate locations and predict regression errors using a CNN. These outputs are used by a PnP algorithm Lepetit et al. (2009) with Ransac Fischler and Bolles (1981) to recover the object pose, demonstrating competitive experimental results at a speed up to 10fps. Instead of regressing 3D coordinate locations, Zakharov et al. (2019) regress UV map locations, which they obtain automatically using spherical projection, and similarly require post-processing with PnP and Ransac to recover the estimated 6DOF pose.

While showcasing impressive numerical and qualitative results, the methods described above require keypoints to be defined for the object in some reference coordinate system, which is arbitrarily chosen for each model. Training data also has to be available for the target object model. Such methods therefore do not generalize to unseen objects. Recent efforts attempt to lift this limitation. Pitteri et al. (2020, 2019) propose to train models which regress translation and rotation invariant local surface embeddings which capture the local geometry of the object’s surface at each pixel location. These discriminative features can be predicted by a neural network from images and computed automatically from a CAD model for unseen objects. The predicted features can be mapped to corresponding candidate 3D object locations and the final pose estimated using a PnP+Ransac algorithm. Their method seamlessly handles symmetric object models. A drastically different solution to this problem is proposed by Xiao et al. (2019). They choose to encode the object model using the PointNet Qi et al. (2017) point cloud encoder, which allows the model to learn a relative transform between the reference input model and the observed posed model in the image. These work present interesting methods which have the potential to generalize to new objects, but require 3D pose annotation data for diverse images and objects, which are tedious to obtain in practice, to achieve their full potential.

Template matching. While feature detection typically works well for textured objects, objects with homogeneous textures and smooth appearance can result in unreliable 2D estimates, leading to erroneous final poses. An alternative approach focuses on directly matching a known object template to visual evidence. Early work in pose estimation matched 2D or 3D planar models represented as edge images to visual evidence. Perkins (1978) propose a complete system to detect the 2D pose, parametrized by 2 translations and one rotation of known objects. The proposed approach extracts groups of curves automatically

from gray-scale images of industrial parts, and produces candidate transformations which best align the model to the extracted curves, taking into account the curve locations and orientations. Similarly, Huttenlocher et al. (1993) detect and recover the 2D rigid pose in the pixel plane of a template model in an binary image. They match the model, also represented by a binary image to edges extracted from the target image. Their matching criterion is based on the Hausdorff distance, which computes the maximal distance of points from the 2D rigidly translated model to the reference image pixels and display results in cluttered scenes. Olson and Huttenlocher (1997) display matching results for 3D object models, such as a tank, by efficiently searching through a hierarchical cluster structure of 2D views of the target object. Their approach estimates translation, rotation and scale by matching edge pixels enriched with orientation information.

Later, Holzer et al. (2009) rely on distance transforms to generate templates given an image representation of an object, and recover an approximate 3D pose by computing the homography between matched reference and target templates. Hinterstoisser et al. (2010) construct a representation based on extracting discretized dominant gradient directions locally from a reference image for improved robustness. However, their method is still susceptible to failure when important clutter is present in the background. Hinterstoisser et al. (2012a) propose to perform template matching with a more robust image representation based on quantized image gradients. For improved robustness to small deformations and translations, each pixel location also stores whether a given gradient direction is present in its neighborhood. This information is efficiently stored in a lookup table using binary strings to enable real-time template matching. A limitation of this line of work is that a large number of templates is required to cover the variety of viewpoints under which a given object can appear. Instead of using image templates as in their previous work (Hinterstoisser et al. (2012a, 2011, 2010)), Hinterstoisser et al. (2012b) rely on a CAD model to generate the target templates, which allows to automatically generate a large number of templates to cover the full range of possible viewpoints.

Sundermeyer et al. (2018) bypass the need for ground-truth 3D information by learning to decode a simple rendered version of an object model from a augmented version to learn features which implicitly capture pose. At inference time, their learnt features can be used to query from a codebook of poses using cosine similarity in the latent feature space.

3D pose regression. Instead of localizing 2D keypoints and lifting these detections given a known object model, several CNN-based methods propose to directly regress 3D informationsuch as 3D vertex locations or object rotation and translation. Su et al. (2015) propose to predict category-specific viewpoints using a mix of real and synthetic data. Kehl et al. (2017) introduce SSD-6D, which builds upon the SSD architecture Liu et al. (2016) and

regress 2D bounding box corners and predict discretized viewpoint and in-plane rotation classes. For improved accuracy, they rely on an edge-based refinement which has to be performed for each candidate object detection. Xiang et al. (2018) use a neural network, PoseCNN, to regress 3D rotations parametrized as quaternions and the center of the object in the image to recover the full 6DOFs of an object model. Their two-step approach brought the total runtime to 10Hz at the time of implementation.

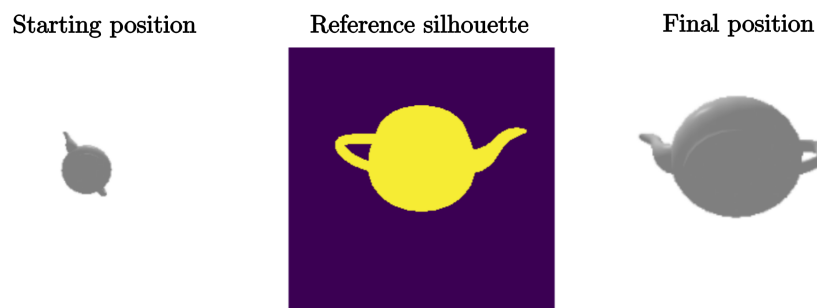
Refining estimated poses. When a coarse initialization is provided, object poses can be further refined for improved accuracy. Differentiable renderers, for instance OpenDR Loper and Black (2014), NMR Kato et al. (2018) and SoftRas Liu et al. (2019), allow to optimize the 6 rigid pose DOFs so that the final silhouette matches a target reference of segmentation. In Figure 2-6, we showcase the converged results of segmentation-based pose optimization for a teapot using differentiable rendering implemented in Pytorch3D Ravi et al. (2020) starting from two different initializations. Such methods can result in accurate final poses provided good initializations 2-6a, but are also prone to converging to poor local minima as we showcase in Figure 2-6b.

Valid initializations obtained by CNN-based methods can also be refined using evidence extracted from the target image. For instance, Kehl et al. (2017) improve their initial predictions using automatically computed edges. However, these refinements have to be computed for each object candidate, increasing the total computational cost.

Instead of refining the object pose with explicit optimization during inference, several methods show the benefits from iteratively refining the object viewpoint using CNNs. Learning to correct the pose has the potential to recover from poor initialization and bypass the need to compute image-level features. A successful paradigm for learnt pose refinement combines a rendering of the object given the current pose estimate and the target image as inputs to a CNNs (Labbé et al. (2020); Li et al. (2018); Manhardt et al. (2018); Zakharov et al. (2019)). This formulation allows the model to correct the current estimate so that the rendering best matches the target image in the pixel domain. While Li et al. (2018) build on PoseCNN Xiang et al. (2018), Manhardt et al. (2018) refines the pose initializations from Kehl et al. (2017). Zakharov et al. (2019) similarly refine the predicted object poses using a learnt model by combining the strengths of the approaches of Li et al. (2018) and Manhardt et al. (2018). Similarly to Li et al. (2018), they predict rotations in the object coordinate system and reuse their loss while starting from an ImageNet Deng et al. (2009) pretrained network as Manhardt et al. (2018). These iterative refinement procedures have shown critical to improve the accuracy of the current best-performing object pose estimation methods for both textured and untextured objects Labbé et al. (2020).



(a) Successful example of camera view estimation using a target silhouette mask and differentiable rendering.



(b) Such iterative methods are strongly dependent on the initialization and often converge to poor local minima.

Figure 2-6: Differentiable rendering allows to optimize the translation and rotation of a target mesh to match silhouette or RGB evidence. Above, we present successful (2-6a) and unsuccessful (2-6b) examples where the object pose is optimized using the Pytorch3D Ravi et al. (2020) tutorial by matching a reference and differentiable silhouette mask starting from two different initial poses.

Estimating the shape of arbitrary objects would allow to model interactions with unknown items. Recently, methods which rely on neural networks have shown impressive results and demonstrate that it is possible to recover functional 3D information from as little as a single RGB image. However, most existing methods require paired images and 3D labels. Such data is available at a scale suitable for training modern neural networks only for a restricted set of object instances and often consists of synthetic renderings. This limits the generalization of trained models to unseen objects and makes them susceptible to synthetic-to-real domain gaps. In Chapter 3, we use a CNN to model the shape of various instances from everyday object categories. We generate a synthetic dataset and use extensive randomization to limit the domain gap. In Chapters 4 and 5, we work under the more restrictive assumption that exact or approximate object models are available and estimate their 3D pose. Using this more restrictive setup, we can directly leverage pixel information to estimate or refine 3D poses during training or fitting.

2.2 Hand modeling

Hand pose estimation from images has attracted research interest since the nineties Heap and Hogg (1996b); Rehg and Kanade (1994), and can generally be categorized into *generative* or *discriminative* methods Erol et al. (2007), or partitioned into its generative and discriminative parts. Generative approaches recover the hand pose estimate by optimizing an explicit hand model to a set of constraints while discriminative methods directly recover hand poses from image data using regression or classification. Generative methods have the advantage of explicitly modelling image evidence as well as priors on the statistical or physical likelihood of hand poses, but are often susceptible to failure in case of inaccurate initializations. Furthermore, optimizing hand models to multiple constraints can imply important computational costs. In contrast, discriminative approaches map the input data directly to a hand pose proposal, and therefore have the potential to provide real-time estimates from as little as a single frame. However, discriminative approaches most often require annotated data, which can be difficult to obtain in practice. This shortcoming often limits the application of discriminative methods to restricted visual and pose domains. Hybrid approaches have been developed to combine positive aspects of generative and discriminative methods. Ballan et al. (2012); Panteleris and Argyros (2017); Taylor et al. (2016); Tzionas et al. (2016) combine generative and discriminative elements such as discriminative keypoint regression followed by hand pose optimization given this evidence. Going further, recent hand datasets are typically annotated using hybrid Hampali et al. (2019); Simon et al. (2017); Tzionas et al. (2016); Zimmermann et al. (2019) methods, and the resulting annotations subsequently used to train discriminative models.

Historically, early methods focused on grayscale images and often required the hand to be initialized in a pre-defined pose at a known location Erol et al. (2007). The availability of commodity RGB-D sensors Kinect (2008); PrimeSense (2013); Shotton et al. (2011) led to significant progress in estimating 3D hand pose given depth or RGB-D input Hamer et al. (2009); Keskin et al. (2012); Moon et al. (2018); Oberweger et al. (2015a,b); Oikonomidis et al. (2011a). We refer to Yuan et al. (2018) for a review of depth-based 3D hand pose estimation. Recently, the community has shifted its focus to RGB-based methods Iqbal et al. (2018); Kulon et al. (2020); Mueller et al. (2018); Panteleris et al. (2018); Simon et al. (2017); Zimmermann and Brox (2017). In the following, we focus on 3D hand pose estimation from color images. We first review pose estimation using articulated hand models 2.2.1. We then present methods which focus on regressing hand keypoints 2.2.2 and finally discuss works that rely on parametric hand models 2.2.3 which model both articulation and deformation.

2.2.1 Articulated pose estimation

A significant portion of hand pose estimation methods rely on tracking articulated hand models, which represent the hand skeleton using rotational joints and model the volumetric extent using rigid primitives Melax et al. (2013); Oikonomidis et al. (2011a); Rehg and Kanade (1994); Tagliasacchi et al. (2015) or Gaussian distributions Sridhar et al. (2013, 2014). More advanced articulated hand models such as Sphere-meshes, which model the hand surface using convex hulls of pair or triplets of spheres Tkach and Tagliasacchi (2016), or articulated triangular meshes based on linear blend skinning (LBS) Jacobson et al. (2014); Lewis et al. (2000) have also been used in the context of hand tracking Ballan et al. (2012); Tzionas (2017).

Rehg and Kanade (1994) track a simplified hand model with 28 DOFs using automatically tracked finger tips and links from grayscale images. They model hand links with cylinders, and track link boundaries in the images to localize the hand parts. Given the simplicity of the visual features they use, their method is sensitive to occlusions and background clutter. Stenger et al. (2001) use a hand model based on truncated quadrics to track the hand pose in grayscale images. While the underlying model has 27 DOFs, their method only tracks 7: the global hand pose and the thumb configuration. de La Gorce et al. (2011) track one or several hands by modeling the texture and pose of an articulated hand model. A parallel line of work investigates single-frame hand pose estimation for articulated models. Matching real images to synthetic renderings of hand pose models was initially explored by Athitsos and Sclaroff (2003), who match automatically extracted contours of a real colored image and synthetic renderings in different characteristic poses to retrieve candidate 3D configurations. Romero et al. (2009) propose to query hand poses from a larger database containing 100k renderings using approximate nearest neighbor search based on HOG features Dalal and Triggs (2005). Similarly to methods which estimate object poses by querying databases, these procedures can estimate hand poses from a single RGB image, but their diversity is limited to the ones present in the generated database.

The more recent work of Panteleris et al. (2018) fits an articulated hand model to noisy 2D joint detections from OpenPose Simon et al. (2017), using the predicted confidence scores to down-weight the discrepancy penalization for less-confident joints. This method allows them to model a wider range of hand poses, but propagates the errors from the underlying 2D keypoint detector.

2.2.2 Keypoint regression

With the advent of CNNs, 3D hand pose estimation has often been treated as predicting 3D positions of *sparse* joints from depth inputs (Ge et al. (2017, 2018); Wan et al. (2018); Yuan

et al. (2018)) or RGB inputs (Iqbal et al. (2018); Mueller et al. (2018); Spurr et al. (2018); Zimmermann and Brox (2017)). Given the success of methods which use convolutional neural networks (CNNs) to perform 2D hand pose Simon et al. (2017) estimation from RGB images, several work lift 2D estimations to 3D Cai et al. (2018); Zimmermann and Brox (2017). Zimmermann and Brox (2017) train a two-stage method, performing first 2D pose estimation and then lifting the pose to 3D, regressing 3D hand joint locations in a hand-centric coordinate frame. Cai et al. (2018) propose an end-to-end trainable architecture for hand pose estimation which predicts 2D keypoint heatmaps as a first stage, 3D keypoints as a second, and finally predicts a depth map from the keypoint locations, which is used for weak supervision when depth data is available. Other methods predict 2D and 3D keypoints jointly. Mueller et al. (2018) use a re-projection layer to jointly regress 2D and root-relative 3D joints. They fit a kinematic skeleton as a post-processing step to obtain a biomechanically valid hand skeleton. Iqbal et al. (2018) train a 3D joint regressor using a $2.5D$ latent heatmap representation that is invariant to depth and scale ambiguities. Spurr et al. (2018) embed RGB images and 3D hand keypoints in a common latent space, which allows them to retrieve plausible hand poses given the input image. Hampali et al. (2019) propose a three-stage architecture which first extracts image features, then 2D heatmaps, which are finally lifted to 3D keypoint locations.

2.2.3 Hand pose and shape estimation

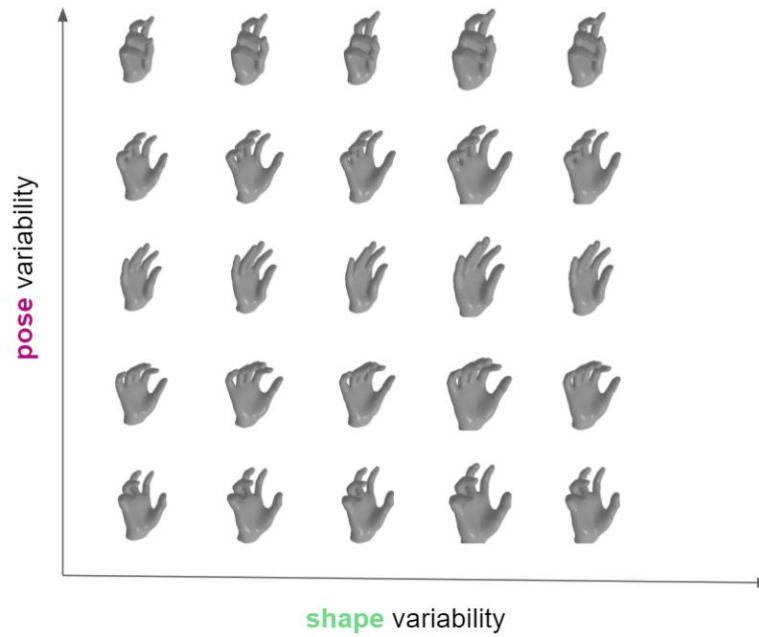
Deformable hand models. Estimating the hand pose by either regressing keypoint locations or using an underlying articulated model can be useful for applications for which a coarse approximation of the human hand is sufficient. However, precisely modeling contacts with objects or with other hands requires millimeter-level accuracy. Several methods have proposed to model the hand surface with higher accuracy than afforded by constrained kinematic models. These methods typically use an underlying mesh with fixed connectivity. Heap and Hogg (1996b) use a hand model based on a Simplex Mesh Delingette (1994), and perform tracking using features computed from detected edges, showing a degree of robustness to background clutter. de La Gorce et al. (2011) allow the bone length of the skeleton of a triangular hand mesh to vary to account for user-specific characteristics. Khamis et al. (2015) propose to learn a hand model based on hand scans. Their model is based on a learnt low-dimensional representation of shape variation which restricts the deformations to be modeled by LBS. Taylor et al. (2016) propose to personalize this hand model in an offline step for each specific user following the method introduced by Tan et al. (2016) before tracking the hand pose.

Recently, an influential approach for hand pose estimation focuses on computing pose and shape parameters of the MANO Romero et al. (2017) hand model. MANO is a para-

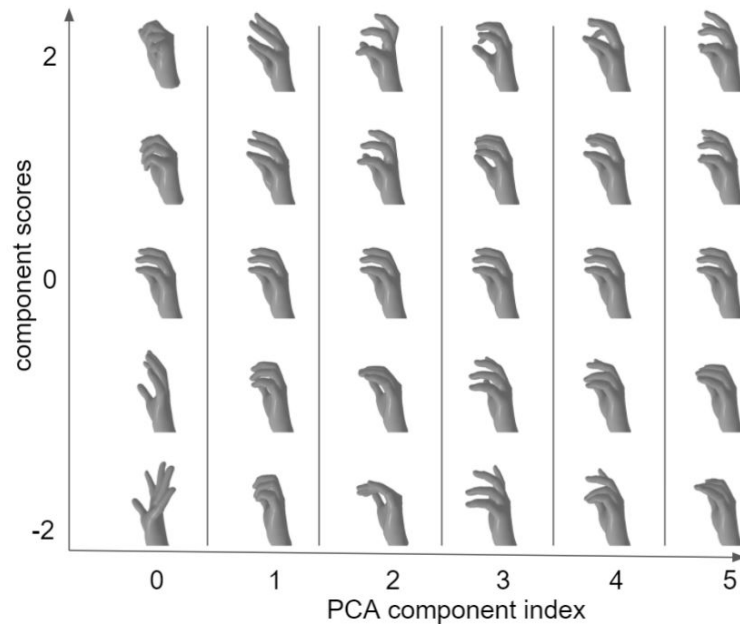
metric model which is obtained by fitting an artist-designed mesh template to hand scans acquired in a multi-view capture setup Romero et al. (2017). MANO disentangles the deformations due to articulated motion and person-specific hand shape. We illustrate in Figure 2-7 the variations in pose and shape space. Our contributions in Chapters 3, 4 and 5 are among the work which rely on MANO for modelling the deformations of the hand surface.

Hand pose dimensional reduction. Statistical analysis of natural grasps have confirmed that a restricted number of degrees of freedom, account for most observed hand pose variations. For instance, Ingram et al. (2008) record the hand pose for 6 subjects, and find that 6 PCA components explained 80% of the variance observed in articulated hand poses in daily activities. More recently, Jarque-Bou et al. (2019) analyze poses captured using a data-glove for a larger population (77 subjects performing 20 grasps) and observe that 12 synergies account for more than 80% of the total hand variation. Several work attempt to capture this low-dimensional articulation and deformation space. Heap and Hogg (1996a) performs Principal Component Analysis (PCA) to find the main variation modes of the hand, modeling the deformations due to the articulations and person-specific deformations jointly. Kendall and Gibbons (1980) reduce the space of articulated hand pose using principal component analysis (PCA). Douvantzis et al. (2013) perform hand pose dimensional reduction in the context of single-frame hand pose estimation. They start from a 20-dimensional articulated pose space and compute principal components. They show that they can capture the training dataset up to a 1mm error with only 10 PCA components. They further propose to handle non-linear relationships between articulations by fitting several linear PCA models across different subsets of the dataset. At inference, they fit the different models using particle swarm optimization (PSO) and keep the solution with the lowest fitting error. MANO applies PCA to the axis-angle pose space of joint rotations. As is illustrated in Figure 2-7b, the two first PCA components capture the global hand flexion-extension and synchronized finger flexion-extension motion respectively.

Regressing hand model parameters. The introduction of the SMPL Loper et al. (2015) body model, which preceded MANO in its definition, was followed by methods which reframed body pose estimation from a single frame as regressing SMPL pose and shape parameters using CNNs Kanazawa et al. (2018a); Pavlakos et al. (2018). These methods rely on the fact that SMPL can be integrated as a differentiable operation in an end-to-end learning framework. Similarly, the introduction of MANO spurred our work presented in Chapters 3 and 4 as well as concurrent efforts which regress mano pose and shape parameters from depth Malik et al. (2018) or RGB inputs Baek et al. (2019); Boukhayma et al.



(a) MANO Romero et al. (2017) maps pose and shape components to the vertex positions of an articulated hand model.



(b) MANO can also be controlled in a low-dimensional pose space, obtained through PCA of the finger's axis-angle rotation space.

Figure 2-7: The MANO Romero et al. (2017) parametric hand model.

(2019). Boukhyama et al. (2019) take predicted 2D keypoint heatmaps as input in addition to the color image and propose to reproject the regressed keypoints to supervise the predicted 3D hand pose with a loss on 2D joint locations. Baek et al. (2019) use differentiable rendering to render the hand silhouette and use a 2D mask loss as an additional source of supervision.

Vertex location regression. Some work offer the promise of arbitrary precision in terms of modeling the hand surface, beyond the current limits of parametric hand models by regressing the locations of hand vertices using GCNs Defferrard et al. (2016). Cai et al. (2018); Kulon et al. (2020, 2019) propose to predict the location of each individual hand vertex from single RGB images. This flexibility comes at the expense of necessary regularization of the object surface during training. Furthermore, annotations at this level of precision are only recently becoming available Smith et al. (2020). Their model describes the surface of the hand, effectively capturing the user specific hand shape. Going beyond a simple articulated model such as the one used by Panteleris et al. (2018), Kulon et al. (2020) recover both the hand shape and pose in world coordinates by fitting the MANO Romero et al. (2017) hand model to 2D detected hand keypoints Simon et al. (2017).

Hand pose and shape estimation from visual data has been explored with various 3D representations. Among them, parametric hand models provide several practical advantages when modeling interactions with objects. Estimating mesh surfaces allows to reason explicitly about interactions and contacts. Furthermore, parametric models have the potential to concisely encode the diversity of hand pose and shape variations in a regularized low-dimensional space. In Chapters 3 and 4 we explore how to best combine the generalization strength of neural-networks and powerful shape priors such as the anthropomorphic MANO model.

2.3 Joint hand-object reconstruction

As we highlighted in the Section 1.1, AR and robotics applications require modelling hands and objects during manipulation. In the following, we present related work which addresses hand modeling during interactions. We first focus in Section 2.3.1 on methods which reconstruct hands from images and model objects to improve the robustness of hand pose estimators. Next, in Section 2.3.2, we discuss methods which specifically model hands and objects together in optimization frameworks. We then discuss in Section 2.3.3 methods which have been used to annotate images illustrating hand-object interactions. Finally, we discuss discriminative methods which reconstruct hand-object interactions from a single RGB frame in Section 2.3.4.

2.3.1 Objects as occluders.

Several work incorporate objects as occluders or context to improve the robustness of hand pose estimation methods. In its simplest form, occlusion modelling suppresses the data which is not directly related to hand pose (Hamer et al. (2009)), while other methods integrate object features to estimate the hand pose (Kjellström et al. (2008); Romero et al. (2010)).

Hamer et al. (2009) track a hand manipulating an object from depth data by modeling hand links with surface patches and enforcing soft anatomic constraints such as proximity between connected links and anatomically feasible joint angles. However, they leverage the manipulated object only to model regions which should be ignored when estimating the hand pose.

Instead of ignoring the object-occluded regions when modelling the hand pose, the occluding object can also be used as a visual cue for grasp recognition. Kjellström et al. (2008) retrieve a hand grasp by querying samples from a database of synthetically rendered images. A test time, they use a single real color image as reference, and obtain corresponding 3D hand-object proposals. To generate the database, they manually pose an articulated hand model in 6 distinctive grasps, and capture images by sampling views in a half sphere, see Figure 2-8a. They extract the hand region based on skin color in the target image, and render the grasped object and background in the synthetic dataset in black, effectively suppressing the background and object from both types of images. They frame grasp recognition as a nearest-neighbor search using HOG Dalal and Triggs (2005) encodings. Removing the object in both images allows their method to model object occlusions in the real and synthetic domains. However, as their method focuses on transferring characteristic human grasps to robotics, they do not model detailed hand poses. Romero et al. (2010) propose a similar approach but query hand poses from a larger database which

contains hand interactions with 33 objects, which we illustrate in Figure 2-8b. Cai et al. (2015) classify hand grasps according to the taxonomy of Feix et al. (2009) from color images using Dalal and Triggs (2005) features or HandHOG features, which weigh HOG features using a skin probability map. They observe improved performance from using the full HOG features, which they attribute to useful information acquired when modeling the manipulated object. Mueller et al. (2017) generate a synthetic dataset of hands with virtual object occlusions, and use this data (see Figure 2-8c) to train two CNNs to detect the hand location and regress 3D joint positions from RGB-D data. Mueller et al. (2018) improve the realism of the generated images using GANs to better generalize to real images. Example images from this dataset are provided in Figure 2-8d. The presence of objects as occluders in the synthetic dataset allows the learnt models to acquire some level of robustness to occlusions during training and to learn a visual grasp prior from the data.

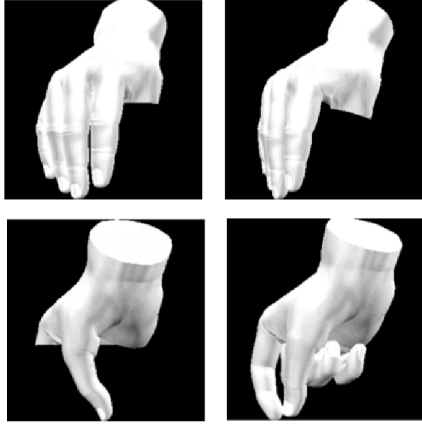
Recently, while most efforts have focused on annotating hands in isolation, some real data annotation efforts have also focused on hands in action. For instance, the Panoptic Studio Dataset Simon et al. (2017) (Figure 2-8e) and FreiHand dataset Zimmermann et al. (2019) (Figure 2-8f) present images of hands holding objects but only annotate hand meshes and keypoints. Learning-based methods which train on these datasets such as Hampali et al. (2019); Kulon et al. (2020); Moon and Lee (2020); Spurr et al. (2020) benefit from this setup and can potentially learn occlusion-robust hand poses.

2.3.2 Joint optimization

Going beyond modelling objects as occluders, several methods propose to jointly model hands and objects during manipulations. A majority of such effort which we detail in Section 2.3.2.1 leverages depth or synchronized multi-view data and tracks hand-object reconstructions in optimization frameworks. Joint reconstruction methods typically enforce explicit or implicitly defined constraints between the hand and the object, which we discuss in Section 2.3.2.2.

2.3.2.1 Model-based tracking

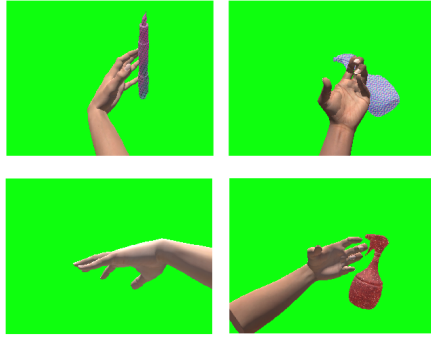
Joint tracking of hands and objects has been studied from single-view RGB-D Hamer et al. (2010); Hampali et al. (2019); Kyriazis and Argyros (2014); Pham et al. (2018); Sridhar et al. (2016); Tsoli and Argyros (2018); Tzionas et al. (2016); Tzionas and Gall (2015), multi-view RGB-D Brahmbhatt et al. (2020); Kwon et al. (2021) as well as multi-view RGB Ballan et al. (2012); Oikonomidis et al. (2011b); Wang et al. (2013) inputs. These generative methods use hand and object models to generate and evaluate pose hypothesis using objective functions which measure how well the poses respects a set of constraints.



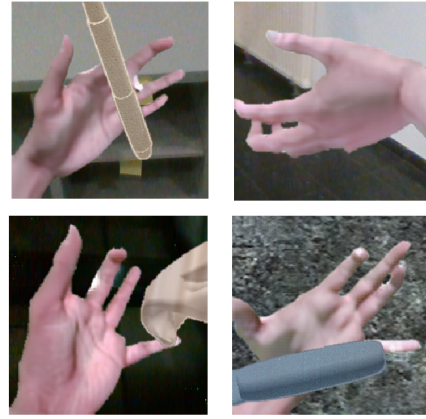
(a) Images from Kjellström et al. (2008)



(b) Images from Romero et al. (2010)



(c) SynthHands dataset images Mueller et al. (2017)



(d) GANerated dataset Mueller et al. (2018)



(e) Panoptic Studio dataset Simon et al. (2017)



(f) FreiHand Dataset Zimmermann et al. (2019)

Figure 2-8: Several hand-object datasets have been proposed to support methods which focus on estimating hand poses and model the occlusions generated by objects.

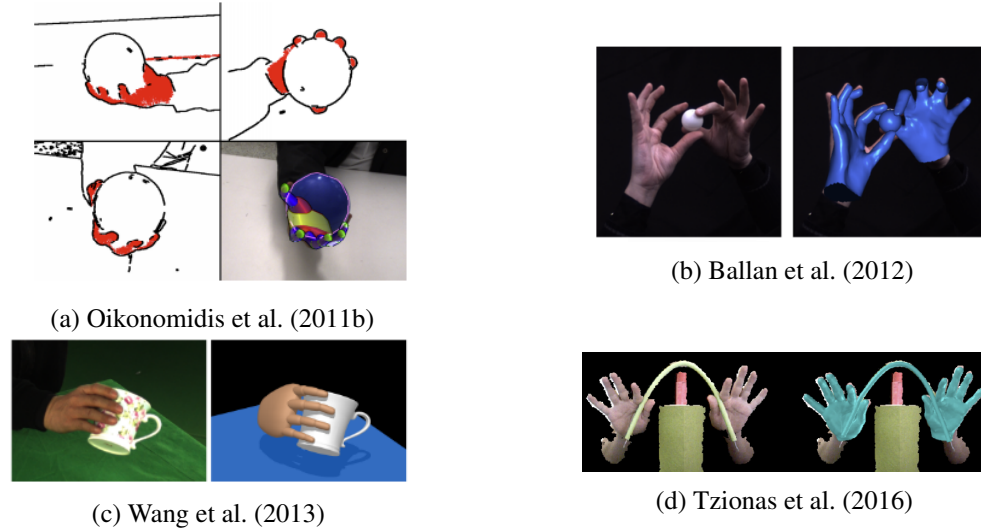


Figure 2-9: Examples of input frames and reconstructed hand-object configurations for model-based tracking methods. All images are reproduced from the original papers.

These constraints can in practice be categorized into data terms, which measure how well the candidate configuration explains the visual evidence and regularization terms, which encode priors on the 3D shapes and poses.

Tracking rigid object manipulations. Joint hand-object tracking has been explored in various optimization frameworks. Hamer et al. (2010) extend the previously introduced hand tracker Hamer et al. (2009) to reconstruct observed grasps, explicitly modelling hand-object contacts. They reconstruct the manipulated object mesh using range data and fit the object model in an offline step using ICP (Besl and McKay (1992)), subsequently recovering the hand pose in a belief propagation framework. Oikonomidis et al. (2011b) estimate the pose of both an articulated hand and simple objects such as cuboids and spheres. They track the hand interacting with an object from detected skin regions Argyros and Lourakis (2004) and Canny edges Canny (1986) in multiple synchronized views using particle swarm optimization (PSO Clerc and Kennedy (2002)). We refer to Figure 2-9a for a visual overview of the multi-view inputs for their method. The hand tracking initialization is facilitated by placing the hand in an approximately known pose, which limits the applicability of the proposed method in practice. Ballan et al. (2012) rely on iterative local optimization using the Levenberg-Marquardt algorithm (Levenberg (1944); Marquardt (1963)) to track strongly interacting hands and objects under limited occlusions. They use optical flow and finger tips detected with discriminative Hough Forests Gall et al. (2011) to guide a LBS hand model. As their iterative optimization method can potentially drift to poor local minima, they explicitly account for outliers and re-initialize the tracking us-

ing simulated annealing when the objective function value crosses a numerical threshold. Wang et al. (2013) first perform kinematic motion tracking and subsequently look for the best motion control with respect to the visual evidence. More specifically, they take into account the hand silhouette and colors as well as edges for the hand and the object. Their optimization relies on interacting simulated annealing (Gall et al. (2006)) to recover noisy pose estimates. Figure 2-9c presents an example of the reconstructions obtained with this method. Sridhar et al. (2016) track interactions between a hand and a simple object model in real-time from RGB-D images. They extend the 2.5D Gaussian Mixture model (Jian and Vemuri (2005)) from Sridhar et al. (2015) to 3D and guide the hand and object pose using the outputs of random-forests Breiman (2001) for hand part and object segmentation. Kyriazis and Argyros (2014) track more challenging scenes presenting two hands interacting with multiple known objects captured by a single static RGB-D camera. They use PSO following Oikonomidis et al. (2012). Going beyond geometric modeling, Pham et al. (2018) predict the forces exerted between the hand and manipulated object in addition to the kinematic configuration solely from visual inputs. To test their method they collect the Manipulation Kinodynamics dataset, which measures the forces during manipulation by equipping the objects with force sensors.

Data annotation by model tracking. Tracking methods can leverage strong cues and have been one of the main resources to recover pseudo-ground-truth 3D information for manipulation images. Ballan et al. (2012); Tzionas et al. (2016) present some of the early results which result in 3D reconstructions which are accurate enough on some sequences to be considered as ground truth. Their method however can only handle limited occlusions, resulting in unnatural motions with spread fingers to avoid these shortcomings, as we illustrate in Figures 2-9b and 2-9d. More recently, Hampali et al. (2019) propose to recover precise hand-object annotations from a single-view RGB-D sequence for known object models from the YCB Çalli et al. (2015) dataset. They rely on learnt components, to segment the object for instance, which are specifically trained for the target object. These predictions are used to guide the joint optimization of the hand and object poses over a sequence of frames. They obtain accurate annotations but rely on smoothness priors which restrict the hand motion speed. Applying their approach therefore requires limiting the motion range for the target sequence. Encouragingly, their work results in one of the first marker-less Hand-Object annotated datasets suitable for training and evaluating learnt methods. Brahmabhatt et al. (2020) propose an approach to annotate synchronized RGB-D images of hand-object interactions. They rely on MoCap to recover very precise poses of 3D-printed objects which they equip with reflecting markers. To estimate the hand pose, they robustly fit the MANO (Romero et al. (2017)) hand model to noisy 2D keypoints pro-

vided by OpenPose Simon et al. (2017) and assume rigid grasps throughout the sequence. This assumption limits the applicability of their method beyond the collected ContactPose dataset. Very recently, Kwon et al. (2021) propose a semi-automatic method based on tracking a scanned model for 8 distinct objects and fitting the MANO Romero et al. (2017) model to multi-view depth data. Their tracking methods sometimes result in erroneous poses, which they filter manually and replace with results from temporal interpolations.

Tracking manipulations of articulated or deformable objects. While most methods focus on rigid object modelling, Tzionas et al. (2016) also reconstruct interactions with articulated objects. Their work extends Ballan et al. (2012), and similarly optimizes a LBS hand model by local minimization of an objective function and assignment of projected hand extremities to detected hand fingertips. Tsoli and Argyros (2018) investigate the interactions between a rigged hand model and deformable surfaces represented as meshes from RGB-D sequences. They minimize an objective function which takes into account hand keypoint detections Simon et al. (2017), SIFT features Lowe (1999) and texture compatibility as well as shape regularization, smoothness and contact priors using the Levenberg-Marquardt method (Levenberg (1944); Marquardt (1963)).

In-hand scanning. While the aforementioned methods assume a known object model is provided, other methods recover the object model directly from the manipulation sequence Panteleris et al. (2015); Rusinkiewicz et al. (2002); Tzionas and Gall (2015); Wang and Hauser (2019); Weise et al. (2011), a process known as in-hand-scanning. While Rusinkiewicz et al. (2002); Wang and Hauser (2019); Weise et al. (2011) discard hand information during reconstruction and rely on distinctive visual object features for reconstruction, Tzionas and Gall (2015) use 3D hand motion explicitly to guide the reconstruction. Their method reconstructs the manipulated object while simultaneously recovering the hand pose from an RGB-D video.

2.3.2.2 Interaction constraints

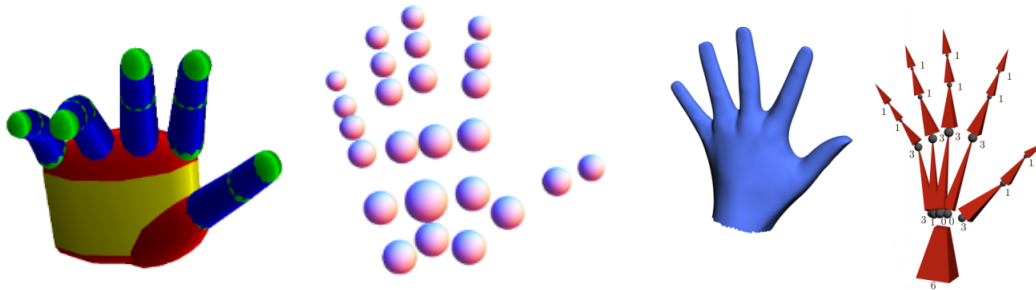
Optimization methods for joint hand-object reconstruction take into account both appearance cues and prior knowledge on the object configurations. In the following we will discuss such hand-object interaction priors. Physical plausibility has received specific interaction, as it provides 3D guidance which can provide useful cues under strong occlusions. In particular, non-interpenetration geometric constraints, which we further describe below are integrated in most objective functions. In Section 2.3.2.2, we further present interaction heuristics which have been used to favor plausible hand-object contacts.

Collision penalization. When reconstructing 3D geometric structures, common-sense rules constrain the space of valid configurations. Specifically, non-interpenetration between solids holds at any point in time and space. This constraint has guided improved pose estimation, in the context of single-person human pose and shape estimation (Bogo et al. (2016); Tzionas et al. (2016)), mutual interactions between people (Jiang et al. (2020)), objects (Kyriazis and Argyros (2014); Oikonomidis et al. (2011b); Tzionas (2017); Zhang et al. (2020)), and/or the environment (Hassan et al. (2019); Kyriazis and Argyros (2013)).

When computing collisions for objects with detailed geometry a trade-off balances speed and precision. Approximating the objects and the hand with simple primitives or more arbitrary convex shapes allows for efficient collision checking (see Gilbert et al. (1988)), while using a more detailed geometry, captured most frequently by a triangle mesh, limits artifacts. We review work that focuses on enforcing non-interpenetration constraints in the context of human reconstruction with a specific emphasis on human-object interactions.

Geometric primitives collision penalization. Modeling complex geometries using simple primitives has the advantage of allowing for fast collision checking. For instance, the collision between two spheres or two cuboids can be checked in constant time. Several hand-object reconstruction methods model the elements in the scene with simple primitives and take advantage of this fact. Hamer et al. (2010) compute the smallest distance between the hand segment and the object mesh in parallel on GPU. They assign a fixed diameter to each hand segment and penalize intersections in their probabilistic framework. Oikonomidis et al. (2011b) couple the tracked hand mesh with a simplified hand collision model based on spheres for efficient collision computations, see Figure 2-10a. For each pair of hand-hand or hand-object primitives, they compute the penetration depth, the minimal distance by which a primitive has to be displaced in order to remove the interpenetration. They handle mutual and self-collision identically, penalizing the maximal penetration depth between all primitive pairs.

Kyriazis and Argyros (2014) also enforce constraints which occur between multiple objects and hands in a scene. Instead of minimizing the maximum penetration depth, they penalize the sum of the penetrations depths across all collision pairs. Kyriazis and Argyros (2013) rely on the collision model of the Bullet simulator Coumans and Bai (2019) to compute collisions between the hand model represented in Figure 2-10a, 3D meshes of the manipulated objects, and the table top. As for Kyriazis and Argyros (2014); Oikonomidis et al. (2011b), the hand collision model is composed of a set of spheres. Similarly to Kyriazis and Argyros (2014), they penalize the sum of penetration depths across pairs of primitives, favoring scenes with limited self and mutual collisions. Relying on simple primitives allows to drastically reduce the computational requirements, but limits the



(a) Hand and collision model of Oikonomidis et al. (2011b)

(b) Hand model used by Ballan et al. (2012)

Figure 2-10: While Oikonomidis et al. (2011b) model collisions by associating collision primitives to their hand model, Ballan et al. (2012) take into account the dense mesh surface and speed-up computations by computing local triangle-triangle intersections. Images reproduced from the original papers (Ballan et al. (2012); Oikonomidis et al. (2011b))

accuracy at which contacts can be modeled.

Local triangle-triangle collision penalization. Partially foregoing efficiency for precision, several work use precise triangular meshes for the hand and object to determine collisions. Given the frequent collision evaluations imposed by iterative optimization frameworks, providing efficient approximations and fast implementations of collision checking is crucial to integrate these terms in practice.

Ballan et al. (2012); Tzionas et al. (2016) use a bounding volume hierarchy to efficiently compute the set of colliding triangles between two interacting hands. To compute the bounding volume hierarchy they rely on Teschner et al. (2004). However, they compute *local* 3D distance fields instead of computing the global 3D signed distance field for each mesh in order to reduce computational complexity. Restricting collisions only to intersecting triangles ignores the part of the colliding mesh which intersects deeper with the collided mesh. For each vertex of a colliding triangle, a repulsion force is applied according to a penalty which depends on the geometry and orientation of the collided triangle.

The same collision loss was reimplemented with GPU acceleration and used in the context of full-body collision checking and penalization in order to prevent self-collision when fitting the SMPL-X parametric human model of Pavlakos et al. (2019a) to 2D evidence. This re-implementation leveraged Karras (2012) for parallelization of BVH computation and show small numerical improvements from the added non-collision constraint, along with qualitative improvements. This collision penalization was also used to constrain human-environment interactions (Hassan et al. (2019)) and human-object interactions, taking into account the full-body scale (Zhang et al. (2020)). Hassan et al. (2019) first scan the surrounding scene, obtaining a pseudo-ground-truth for the static environment, and enforce interpenetration penalization to limit the range of possible human poses. They show quan-

titative and qualitative improvements in terms of reconstructed human meshes. Zhang et al. (2020) add the same collision loss to limit collisions between the human and interacting objects in the context of scenes with multiple persons and objects. They rely on a user study to qualitatively assess the contribution of each of the terms they introduce to regularize their joint scene optimization. The study results in inconclusive numerical improvements from the added collision term while resulting in important qualitative improvements.

Global signed-distance field collision penalization. Precise collision penalization for watertight meshes was implemented by Jiang et al. (2020) with GPU acceleration to minimize the penetration depth in the context of multiple person pose and shape estimation. For this purpose, they compute a per-person signed distance field and penalize colliding vertices using a robust differentiable loss. They store SDM values in a grid and use trilinear interpolation to differentiably penalize colliding vertices. They penalize the sum of penetration depths, using the Geman-McClure robust error function Geman and McClure (1987). As their computation is based on computing ray-triangle intersections Akenine-Möller and Trumbore (1997), collisions can not be resolved for non-watertight meshes.

Close vertex heuristics. In order to physically constrain the hand-object reconstructions of the HO-3D dataset, Hampali et al. (2020) minimize the projection of the vector between the penetrated hand vertex and the closest object vertex on the normal of the given object vertex. They only take into account vectors which have a positive scalar product with the object normal, which allows to penalize only penetrating hand vertices. A similar approach is taken by Brahmabhatt et al. (2019). They manually or automatically extract contact regions on the object surface and apply a repulsive term to the remaining points, which push the hand vertices away.

Interaction heuristics In addition to collision constraints, several work propose to introduce additional interaction heuristics which stem from statistical or intuitive reasoning on the space of plausible grasp configurations. Rijpkema and Girard (1991) present an interactive method to generate natural-looking grasps for primitive object shapes. They rely on human grasp heuristics such as the fact that the thumb is almost always in contact with the object, and that grasps most often involve contacts on opposite object faces. In order to generate plausible contacts, they start from a feasible initial hand position and interpolate the finger tips of the grasping fingers along the line which links them until the fingers collide with the target object. Their method is not suited for some specific grasps, for instance grasping a mug by the handle, which do not involve contact at the thumb’s tip. Hamer et al. (2010) learn a distribution over hand segments from their tracked data. They train the hand pose prior in the object coordinate frame, using manually annotated keypoints in key frames, and use the prior at inference time to favor plausible grasps. Wang et al. (2013)

enumerate contacts between hand parts and the object below a given distance threshold in order to bias the sampled poses towards configurations where the hand produce forces and torques on the manipulated objects. Tsoli and Argyros (2018) similarly enumerate all possible finger-tip contact/no-contact configurations and keep the one that best explains the visual evidence in the interaction region.

Physics constraints. Physics simulations allow to model complex composite scenes. For instance, given an RGB image as input, Gupta et al. (2010) progressively build urban scenes from the ground plane using simple cuboid primitives and enforce physics constraints such as static equilibrium and the existence of support forces. Physics simulation engines allow to go beyond the modeling of static scenes by reconstructing feasible hand and object *trajectories*. By taking into account the dynamics of the scene, methods can produce plausible interactions where the object motion is explained by the hand displacements. Constraining the outputs to be valid in simulation forces the outputs to obey physical rules defined in the simulation engine, and often results in qualitative improvements of the reconstructed trajectories which can also correlate with quantitative improvements.

Several work reconstruct hand motions which explain the displacements of the object in simulation starting from either visual inputs (Kyriazis and Argyros (2013); Wang et al. (2013)) or MoCap data (Ye and Liu (2012)). They define energy functions on the scene by measuring discrepancies between the proposed hand motion and visual evidence and iteratively advance the state of the scene starting from the previously computed state. Starting from body and object motion capture data, Ye and Liu (2012) sample a set of diverse fine-grained hand manipulation motions which explain the observations. They generate contact point trajectories and forces which drive the object motion, resulting in temporally coherent gestures. Kyriazis and Argyros (2013) start from a sequence of RGB-D images and attribute all object motion to the hand, their "single-actor hypothesis". In the simulation, objects are endowed with mass and surface friction, and the hand is the only active object. They simulate the physics using the Bullet Coumans (2013) engine and search for the best hand pose using PSO. They initialize the hand pose in each frame in the vicinity of the pose computed in the previous one. Computed collisions allow to estimate the restitution force and advance the state of the scene given the current hypotheses. The most plausible hand pose, which leads to the evolution of the scene which best coincides with the observed changes is selected during tracking. Wang et al. (2013) rely on the Open Dynamics Engine Smith (2008) to generate physically plausible hand motions from RGB videos. They reconstruct dexterous object manipulations including precision grasps by sampling valid poses for the hand and object in the vicinity of tracked trajectories. However, their method relies on constraints which they impose on the environment to facilitate hand and object

segmentation. They use a simple green background and an initial pose where the hand is in extended flat position and away from the object.

Using physical constraints can result in improved trajectories, as they rely on contacts during manipulation and enforce temporally coherent trajectories. However, these methods come with the overhead of interfacing a simulator and inherit their limitations. The actual physical plausibility of the trajectories is limited by the realism of the simulator, which typically approximate real-world physics with discretized time steps.

2.3.3 Hand-object 3D annotations.

Recent efforts to obtain accurate annotations for hand-object interactions most often still require manual steps for preparation, annotation or verification. In Section 2.3.2.1, we detailed fitting-based approaches for hybrid 3D hand-object annotation which allowed to collect the Hands in Action Tzionas (2017), HO-3D Hampali et al. (2019), ContactPose Brahmbhatt et al. (2020) and H2O Kwon et al. (2021) datasets. While these datasets were annotated using multi-view or depth information, a color stream is also available. As we previously mentioned, these methods are difficult to scale to more diverse datasets in practice.

Other work propose to annotate datasets using sensors which are visible in the color stream. Kry and Pai (2006) use both MoCap sensors for the hand and force sensors at the finger tips to estimate the pose of the hand and object during manipulation. Pham et al. (2018) collect the Kinodynamics dataset to evaluate their method. In order to measure forces, they equip objects with sensors which measure accelerations in direction and orientation as well as force sensors. While Pham et al. (2018) require the hand grasp to be fixed, Garcia-Hernando et al. (2018) can capture natural hand-object interactions. Garcia-Hernando et al. (2018) introduce the FPHAB dataset, which is annotated using magnetic sensors taped to the hand and the object. The sensors in FPHAB on the hands are clearly visible. Although the objects sensor are not visible, as they are taped inside each item, small deformations to the object’s surface can skew the pose measurements. Furthermore, not all objects can be easily rigged with concealed sensors. In addition to corrupting the images, equipping hands and objects with sensors incurs a preparation step to equip the hands and objects which limits the scalability of these approaches.

Chao et al. (2021) scale the annotation effort to 8 RGB-D synchronized cameras and 20 objects from the YCB dataset, creating the DexYCB dataset, one of the largest hand-object datasets currently available. They **manually** annotate 2D locations for hand and object keypoints, and fit the pose of the known YCB object (Chao et al. (2021)) and MANO (Romero et al. (2017)) models to these detections. This procedure allows them to allow for unrestricted hand poses, which thus represent natural and spontaneous grasps. Their

method relies on a specific hardware setup, crowd-sourced manual annotations, and known textured object models, which limits its scalability and generalization.

2.3.4 Hand-object reconstruction from a single RGB frame

While fitting methods typically rely on tracking and leverage temporal context to recover accurate hand poses during manipulation, discriminative methods can produce plausible estimates starting from only a single RGB image. While the pioneering work of Romero et al. (2009) focuses on hand poses, they recover a candidate grasp along with a possible object from a database, and can be seen as one of the first work to perform joint hand-object reconstruction from a monocular color image. The quality of retrieved candidate grasps is intrinsically limited by the diversity of object models the database, and retrieving an accurate object model assumes that the given model is available in the database.

As discussed in Section 2.1.1, objects come in many shapes. While curating a database of all possible hand poses could be considered, generating a similar database with arbitrary objects raises practical issues. Discriminative methods therefore focus on either predicting the pose of known objects, as we highlight in Section 2.3.4, or attempt to reconstruct the target object from the image, as we describe in Section 2.3.4. Most recent advances have approached the hand-object joint reconstruction problem through a learning perspective, using CNNs to extract learnt features from RGB images and regress hand and object poses and shapes Lepetit (2020).

Grasp pose estimation for known objects. Assuming the object model is known allows to significantly constrain the manipulation reconstruction problem, which can be cast as the pose estimation problem, which is described in a space of lower dimension. Single-frame methods have focused on rigid objects and parametric hand models, effectively limiting the dimension of the search space.

Tekin et al. (2019) extend the YOLO framework (Redmon et al. (2016)) to both detect and predict the hand and object poses in a single forward pass. They showcase real-time results, and demonstrate improvements from additional temporal modelling using a Long Short-Term Memory (LSTM) network. While their method can theoretically detect several object and hands, in practice, their learnt method is limited by the restricted training domain of the FPHAB (Garcia-Hernando et al. (2018)) and Dexter+Object (Sridhar et al. (2016)) datasets. Doosti et al. (2020) predict 2D locations for the hand joints and object bounding box corners, which they subsequently lift to 3D using a GCN. Interestingly, they note that for the FPHAB dataset, which is annotated using magnetic sensors, roughly half of the annotated frames have a keypoint which projects outside of the pixel image boundaries. Following this observation, they argue against methods which start by detecting the hand

before estimating the keypoint and adopt a direct regression approach. Their architecture is based on a simple ResNet10 He et al. (2015) backbone for 2D keypoint regression, and subsequent refinement and lifting to 3D using graph convolutions. They supervise the learning both with 2D losses on the projected keypoints and a loss on the 3D coordinates. They demonstrate real-time performance and improved results compared to Tekin et al. (2019) on the FPHAB dataset.

Reconstructing objects and grasps. While all the methods presented in Section 2.3.4 assume a known object model is provided, several recent approaches simultaneously estimate the object and hand shape from the input image. These methods have to solve for the ill-posed object shape estimation task along with the hand pose estimation problem. Methods which tackle this arguably more challenging task sometimes work in a normalized space, centering the reconstruction on the hand root (as we do in Chapter 3) and additionally scaling the reconstruction to a normalized scale (Karunratanakul et al. (2020)).

Kokic et al. (2019) focus on a restricted set of object categories: bottles, mugs, knives and bowls. They retrieve an approximate object model and estimate its pose along with the global translation, rotation and 20 joint orientations for an articulated hand model from a single frame. They generate a synthetic dataset to train their retrieval and pose estimation CNN using GraspIt! Miller and Allen (2004) a software which automatically generates force-closure grasps for arbitrary object meshes, which we also use in Chapter 3 for a similar purpose. In order to bridge the domain gap, they use Cycle-GAN Zhu et al. (2017) to improve the realism of the rendered images. They refine the regressed hand pose in the GraspIt! software, first correcting for the object pose by keeping the hand pose fixed and subsequently using the automatic grasping procedure to recover a configuration without interpenetration and with valid surface contacts. Their method can only retrieve meshes from a fixed set of object models in the database by encoding the object shapes and appearance in the same embedding space.

Karunratanakul et al. (2020) propose to reconstruct the object shape and hand surface by learning to predict for each 3D point in a normalized coordinate system the distance to both the hand and object surface. This representation is directly recovered from the input image and allows to recover surfaces using simple post-processing and a corresponding MANO model by fitting the parametric hand model to the predicted distance field. Their method can reconstruct arbitrary object shapes, with no prior restriction on object category or topology. Similarly to joint hand-object pose estimation methods described in Section 2.3.4, the main shortcoming of this method comes from the limitations of available datasets. Currently, most of their experiments are conducted using our synthetic dataset presented in Section 3.2. As it is learning-based, it would benefit from additional annotated

images to generalize to a larger variety of objects and grasps.

We reviewed the literature on hands and object modeling during interactions. Most previous methods use known models, tracking and work in constrained capture setups. We focus on the challenging scenario where only color images are available. Neural networks can make meaningful predictions under this constraint by learning strong priors from training data. In Chapters 3 and 4, we train models for the task of recovering dense 3D representations of hands and manipulated objects from a single RGB frame or video clip. Following previous work which target plausible object manipulation sequences, we explore in Chapters 3 and 5 how interaction constraints can improve reconstruction in learning and fitting frameworks.

Chapter 3

Learning joint hand-object reconstruction from synthetic grasps

In Section 1.2.3, we underlined that most hand-object interactions are recorded using standard color cameras, and present unavoidable occlusions. We want to develop methods which estimate the 3D geometry of hands and objects from RGB images so that we can analyze how humans grasp everyday objects. We are interested in approaches which are widely applicable and focus on a single color image as input and unknown manipulated objects. We develop a learning-based model to benefit from the successes of neural networks, and predict the object shape and associated hand pose. In this chapter, we present how we automatically generate synthetic training data (Section 3.2) and develop a CNN architecture (Section 3.3) to predict plausible grasps in a single forward pass.

While manual annotations catalyzed progress in 2D recognition and detection tasks, 3D annotations are typically tedious to acquire and incur important annotation costs as we detailed in Section 2.3.3. Data collection efforts have resulted in diverse images with image-level annotations for the ImageNet Deng et al. (2009) dataset and pixel-level details for the COCO Lin et al. (2014) and PASCAL Everingham et al. (2015) datasets. Such datasets have allowed to train CNN-based methods which demonstrate impressive performances when compared with earlier methods as well as in-the-wild generalization capabilities for image classification (ResNets He et al. (2015)), detection (Fast-RCNN Girshick (2015), Faster-RCNN Ren et al. (2015)) and segmentation (Mask-RCNN He et al. (2017); Kirillov et al. (2019)). Datasets with 3D annotations have been collected in more restricted settings for the tasks of object Hinterstoisser et al. (2012b); Hodan et al. (2017, 2020); Sun et al. (2018), human Hanbyul Joo and Sheikh (2015); Ionescu et al. (2014) and hand pose estimation Simon et al. (2017); Zimmermann et al. (2019). Annotation of these datasets often require complex multi-view setups (Hinterstoisser et al. (2012b); Ionescu et al. (2014); Simon et al. (2017); Zimmermann et al. (2019)) and some level of manual processing (Simon

et al. (2017); Sun et al. (2018); Zimmermann et al. (2019)). Up to 2018, available hand-object datasets with 3D annotations were either too small for training neural network (Pham et al. (2018); Tzionas (2017)), had missing annotations (Sridhar et al. (2016)), or visible markers which contaminate the appearance of the RGB frames (FPHAB Garcia-Hernando et al. (2018)). Data scarcity limited the development of CNN methods for 3D hand-object pose estimation.

Programmatically generating synthetic data provides a compelling alternative to complex hardware setups and manual annotations. The development of powerful engines allows to generate a larger number of annotated images than afforded by manual annotations. Simultaneously, the progress towards photo-realistic rendering of 3D scenes facilitates generalization to the real domain. We argue that generating synthetic data is an interesting approach to provide training data for hand-object reconstruction and harness the potential of CNNs for the challenging grasp reconstruction task.

In this chapter, we focus on reconstructing grasps from a single color image. We first review related work on synthetic data generation and automatic grasp generation in Section 3.1. We then present several contributions in automatic reconstruction of hand-object interactions from RGB images. First, we describe our automatic approach to generate and render synthetic data in Section 3.2. We create ObMan, a new large-scale synthetic dataset of hand object interactions. Second, we propose one of the first architectures which jointly regresses hand and object shapes from a single RGB frame in Section 3.3. Our end-to-end differentiable method allows to integrate physical constraints at train time, and results in improved interaction modeling. We present experimental results which validate our contributions in Section 3.4, and conclusions in Section 3.5.

3.1 Related work

We review prior work on synthetic data generation in Section 3.1.1 and provide pointers to reviews for automatic grasp generation using robotic grasping methods and software in Section 3.1.2.

3.1.1 Synthetic data rendering

Synthetic data was initially used in the context of computer vision to target low-level vision tasks such as disparity estimation and optical flow prediction Barron et al. (1994); Little and Verri (1989) and has become increasingly important with the emergence of data-hungry algorithms such as CNNs (Butler et al. (2012); Ilg et al. (2017); Peris et al. (2012)). While some datasets have been repurposed from the gaming and movie industry, increasingly,

synthetic datasets have been generated specifically to circumvent the lack of real annotated data. Efforts to facilitate the generation of synthetic data, such as plugins for existing gaming engines Weichao Qiu (2017) have facilitated such initiatives. We refer to Nikolenko (2019) for a general overview of the use of synthetic dataset to train deep learning methods and discuss the use of synthetic data for 3D pose and shape estimation in the following.

Synthetic data for pose estimation. Generating images through rendering generates pixel-level accurate annotations, as rendering allows to directly generate an image from a ground truth 3D scene. In many cases, it is the only way to guarantee such a level of accuracy for 3D annotations. Rendered synthetic data has been a cornerstone of the development of learning-based pose estimation methods from RGB frames for rigid (Labbé et al. (2020); Loing et al. (2018); Su et al. (2015); Xiang et al. (2018)) and articulated (Labbé et al. (2021); Lambrecht and Kästner (2019)) objects as well as humans (Ionescu et al. (2014); Varol et al. (2017)) and hands (Mueller et al. (2018, 2017); Zimmermann and Brox (2017)).

Synthetic data for object pose estimation. Su et al. (2015) render a synthetic dataset for the task of class-dependent object viewpoint estimation, and demonstrate that their synthetic data positively contributes to learning more robust features for this task. Xiang et al. (2018) complement the real YCB-Video dataset with synthetic renderings and Loing et al. (2018) generate synthetic images for robot-relative object pose estimation from uncalibrated cameras. The recent BOP challenge for 6D object pose estimation Hodan et al. (2020) generates realistic synthetic images, closely following the method presented in Hodaň et al. (2019), for each of the 7 target object pose estimation datasets. Domain randomization, the process of applying extensive augmentation to synthetic data is critical for generalization to real data Tobin et al. (2017). Training solely on synthetic data can results in important performance drops Kehl et al. (2017); Zakharov et al. (2019) compared to using real data. However, with proper data augmentation and methods, learning from synthetic datasets results in competitive performances Labbé et al. (2020).

Synthetic data for object shape reconstruction. The synthetic images associated with ShapeNet Chang et al. (2015) and ModelNet Wu et al. (2015) are the main training data resources for learning-based shape reconstruction methods such as the one presented by Choy et al. (2016); Groueix et al. (2018b); Mescheder et al. (2019); Qi et al. (2017); Wang et al. (2018). When training solely on these synthetic datasets, the domain gap provides some generalization to real images. However, existing methods do not directly generalize for more challenging images, for instance when they present objects which are more difficult to segment from their background. In cases when shape diversity is restricted and larger real datasets are available, for instance in the case of clothed humans, promising results have been presented Saito et al. (2019).

Synthetic data for human pose estimation. Several work which recover 3D human information from RGB images have proposed to generate synthetic data with computer graphics software for training purposes. Chen et al. (2015) generate a synthetic dataset based on the SCAPE human model Anguelov et al. (2005), varying poses, clothing and backgrounds. They show that their data can be combined with real data to improve CNN-based 3D pose estimation models. Varol et al. (2017) generate a larger scale dataset of posed humans, using the SMPL model Loper et al. (2015) and MoCap data Carnegie Mellon University (2001) to render images with improved realism compared to Chen et al. (2015). They demonstrate that this data can be used to learn depth, part segmentation and later full 3D human shapes Varol et al. (2018). Zimmermann and Brox (2017) generate a synthetic dataset of humans with articulated hand motions to train the first CNN which regresses 3D hand keypoints directly from RGB images. Mueller et al. (2017) render a dataset with randomized object occlusions to train their hand pose estimator. Mueller et al. (2018) show that mixing synthetic data post-processed with Cycle-GAN Zhu et al. (2017) to match real image statistics with synthetic data improves the pose estimation performance.

3.1.2 Automatic grasp generation

Methods which provide 3D data from pose estimation for humans in isolation from their environment use MoCap to provide 3D poses. Such data is difficult to collect in practice for hand-object interactions, as MoCap often relies on equipping the hand and objects with markers which are visible and sometimes restrict the range of possible motion as for the ContactPose (Brahmbhatt et al. (2020)), Manipulation Kinodynamics (Pham et al. (2018)) and FPHAB dataset (Garcia-Hernando et al. (2018)) datasets.

Alternative paths can be taken to generate 3D data of anthropomorphic object grasps. In the following, we discuss methods which propose to synthesize 3D hand models using heuristics and analytic measures in Section 3.1.2 and data-driven methods for grasp generation in Section 3.1.2.

Analytic grasp generation. Several methods propose to generate grasps for anthropomorphic hands by devising grasping strategies based on heuristics and evaluate the quality of the generated configurations using analytic metrics. These methods typically assume rigid contacts with friction, compute the grasp wrench space (GWS), the space of wrenches which can be resisted by the grasp (Ferrari and Canny (1992)). Most methods evaluate the grasps by computing how well they can resist a set of forces and torques. A good overview of automatic methods for grasp synthesis is provided by Sahbani et al. (2012).

Data-driven anthropomorphic grasp generation. While automatic grasp generation using heuristic approaches for sampling and analytic measurements for evaluation are scalable, they only partially correlate with successful grasps in the real world as pointed out by Balasubramanian et al. (2012). Learnt grasp synthesis from data provides an interesting way to leverage potentially more reliable priors for grasp generation. We refer to Bohg et al. (2014) for a review of data-driven grasp synthesis and detail recent deep-learning based methods for grasp generation using hand models. Corona et al. (2020) learn to generate hand poses which represent candidate grasps for 58 YCB (Çalli et al. (2015)) objects from a RGB image representing the target objects. They maximize grasp metrics during training using the loss we introduce in Section 3.6 along with a GAN loss and an additional term which penalizes grasps below the table plane. Karunratanakul et al. (2020); Taheri et al. (2020) propose to generate grasps using a Variational Auto-Encoders (VAEs Kingma and Welling (2014)) by regressing MANO Romero et al. (2017) parameters for Taheri et al. (2020) and predicting implicit grasping fields which encode the distance to the hand and the object at 3D coordinate locations for Karunratanakul et al. (2020). Jiang et al. (2021) first predict grasps given an object model and subsequently estimate contact locations based on the grasp and input object shape to obtain improved grasps through fitting.

3.2 Generating hand-object interactions

To overcome the lack of adequate training data for our models, we generate a large-scale synthetic image dataset of hands grasping objects which we call the *ObMan* dataset. Here, we describe how we scale automatic generation of hand-object images. We first detail the steps we take to automatically construct the 3D grasps in section 3.2.1, and then explain the rendering process in section 3.2.2

3.2.1 Automatic grasp generation

We strive to propose a method for automatic generation of grasps for arbitrary objects. In the following, we describe how we select the objects for our dataset and automatically generate a large number of grasps for each of the selected objects. We further describe our procedure to sort and filter them in order to select plausible grasps.

Objects. In order to find a variety of high-quality meshes of frequently manipulated everyday objects, we selected models from the ShapeNet Chang et al. (2015) dataset. As illustrated in Figure 3-1, we selected 8 object categories of everyday objects: bottles, bowls, cans, jars, knives, cellphones, cameras and remote controls. We showcase the diversity of

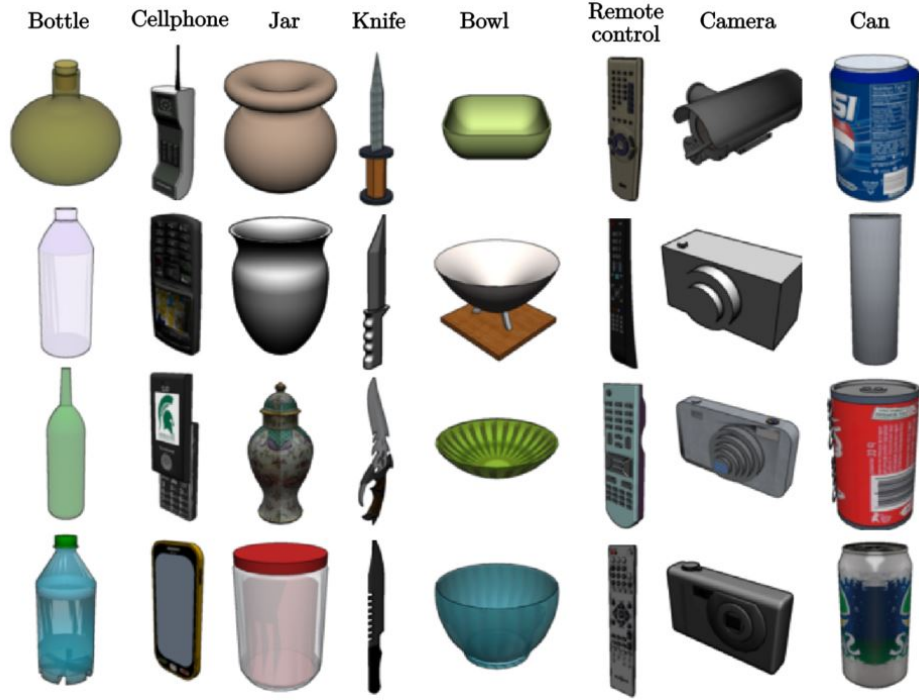


Figure 3-1: We select objects from 8 graspable object categories from the ShapeNet Chang et al. (2015) database.

the object shape categories on the *jar* category in Figure 3-2. Objects in the ShapeNet database can be of arbitrary topology and mirror the natural diversity of object shapes. In total, we start with 2772 meshes split among the training, validation and test sets.

Automatic grasping. In order to generate plausible grasps, we use the GraspIt software Miller and Allen (2004) following the methods used to collect the Grasp Database Goldfeder et al. (2009). In the robotics community, this dataset has remained valuable over many years Sahbani et al. (2012) and is still a reference for the fast synthesis of grasps given known object models Lenz et al. (2015); Mahler et al. (2017). We favor simplicity and robustness of the grasp generation over the accuracy of the underlying model. The software expects a rigid articulated model of the hand. We transform MANO by separating it into 16 rigid parts, 3 parts for the phalanges of each finger, and one for the hand palm. Given an object mesh, GraspIt produces different grasps from various initializations. Following Goldfeder et al. (2009), our generated grasps optimize for the grasp metric but do not necessarily reflect the statistical distribution of human grasps. We present a selection of object grasps generated by the GraspIt software in Figure 3-3

Grasp quality computation and sorting. GraspIt generates a large variety of grasps by exploring different initial hand poses. However, some initializations do not produce good

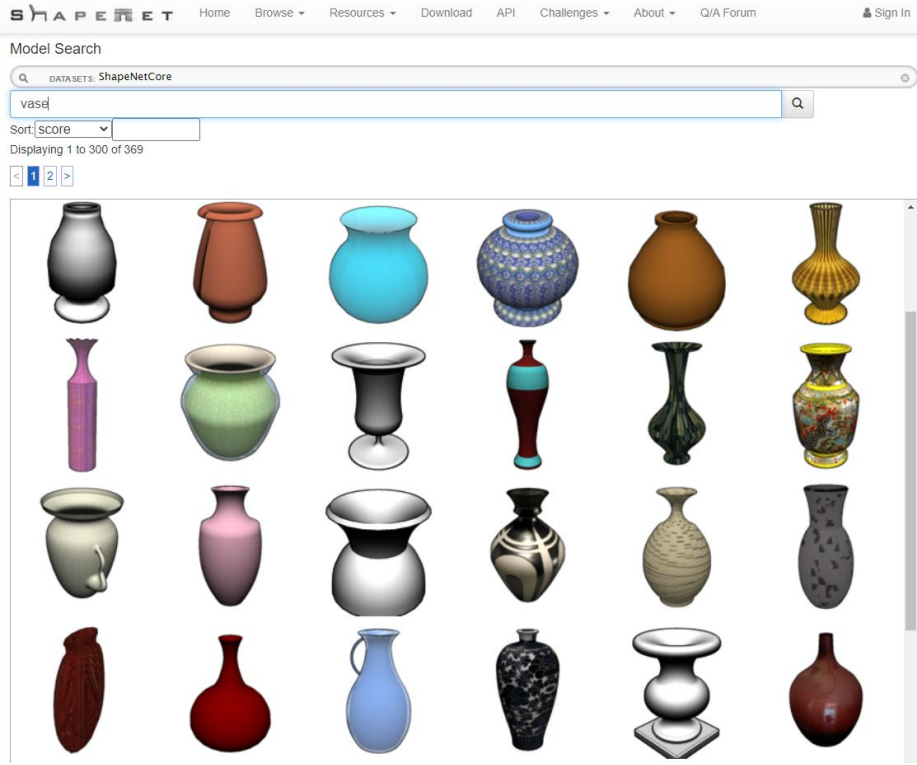


Figure 3-2: We select objects from the ShapeNet Chang et al. (2015) database. We present here random examples of objects from the vase category which we use for our ObMan dataset in the ShapeNet model exploration interface <https://shapenet.org/model-querier>. Objects in ShapeNet can be composed of an arbitrary number of parts and present different topologies.

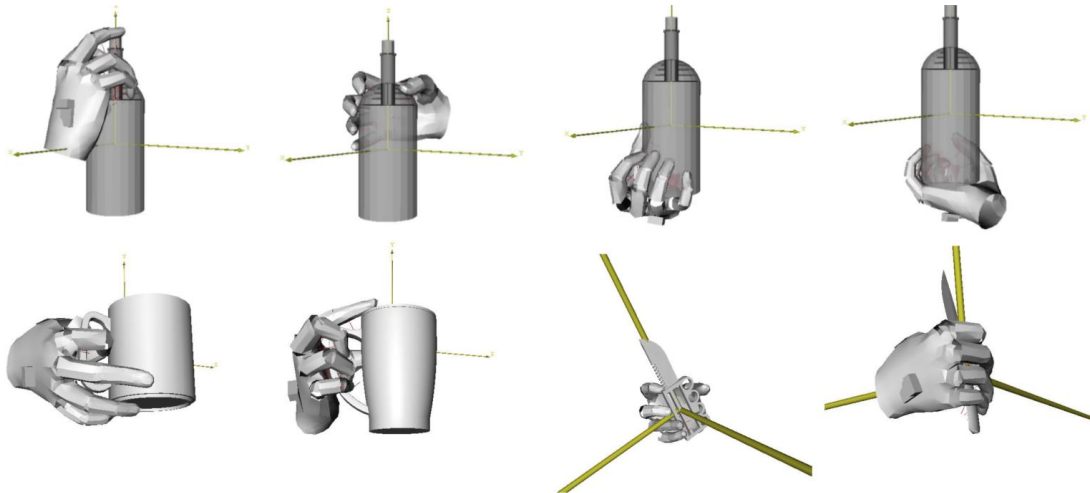


Figure 3-3: We present different grasps generated by GraspIt Miller and Allen (2004) for a single object CAD model as described in Section 3.2.1 in the first row, and grasps for 4 different ShapeNet Chang et al. (2015) models in the second row.

grasps. Similarly to Goldfeder et al. (2009) 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 Ferrari and Canny (1992). Each grasp produced by GraspIt Miller and Allen (2004) defines contact points between the hand and the object, which allows to compute the GWS. The GWS 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 need to project the grasp quality on a single dimension in order to perform automatic sorting. Following the sorting, we can define a cutoff threshold for the defined metric, which allows us to filter a set of grasps which we keep for the final rendering step explained in Section 3.2.2. 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 Feix et al. (2016) 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:

$$G = \gamma_{palm} \sqrt{N_p} \|\varepsilon, v\|_2. \quad (3.1)$$

We find that keeping the two best grasps for each object produces diverse grasps of good quality and generate a total of 21K grasps.

Body pose. For realism, we render the hand and the full body (see Figure 3-6). The pose of the hand is transferred to hands of the SMPL+H Romero et al. (2017) model which integrates MANO to the SMPL Loper et al. (2015) statistical body model, allowing us to render realistic images of embodied hands. Although we zoom our cameras to focus on the hands, we vary the body poses to provide natural occlusions and coherent backgrounds. Body poses and shapes are varied by sampling from the same distribution as in SURREAL Varol

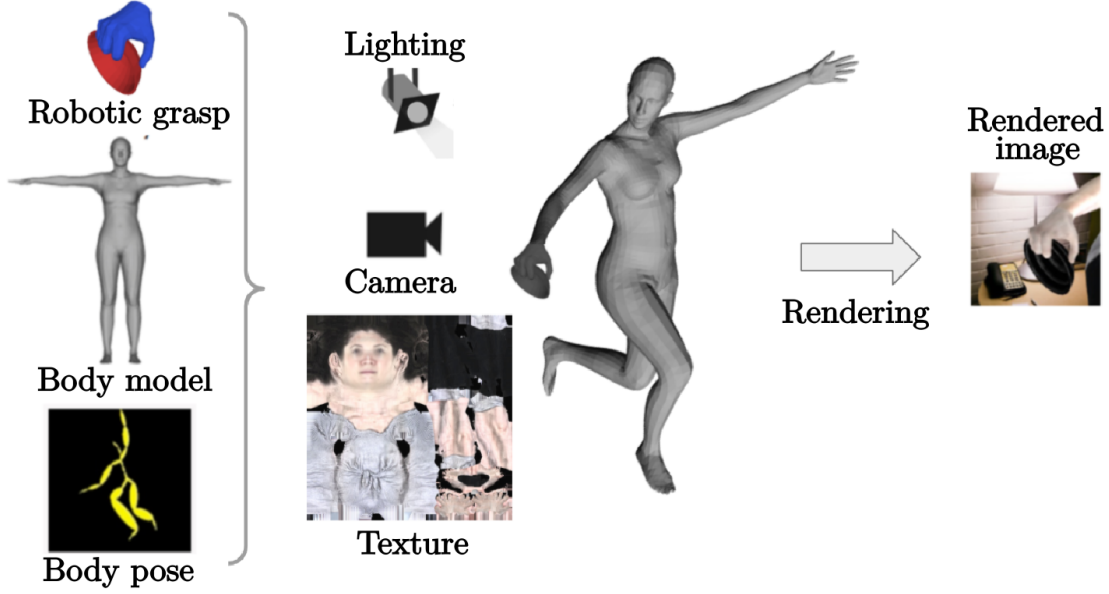


Figure 3-4: We render a full-body posed human model grasping an object with realistic textures obtained by combining body and hand textures .

et al. (2017); i.e., sampling poses from the CMU MoCap database Carnegie Mellon University (2001) and shapes from CAESAR Robinette et al. (2002). In order to maximize the viewpoint variability, a global rotation uniformly sampled in $SO(3)$ is also applied to the body. We translate the hand root joint to the camera’s optical axis. The distance to the camera is sampled uniformly between 50 and 80cm.

3.2.2 Grasp rendering

We build upon the rendering method of Varol et al. (2017) to render our images of hands grasping objects. An illustration of the rendering procedure is presented in Figure 3-4.

Textures. Object textures are randomly sampled from the texture maps provided with ShapeNet Chang et al. (2015) models. The body textures are obtained from the full body scans used in SURREAL Varol et al. (2017). Most of the scans have missing color values in the hand region. We therefore combine the body textures with 176 high resolution textures obtained from hand scans from 20 subjects. The hand textures are split so that textures from 14 subjects are used for training and 3 for test and validation sets. For each body texture, the skin tone of the hand is matched to the subject’s face color. Based on the face skin color, we query in the HSV color space the 3 closest hand texture matches. We further shift the HSV channels of the hand to better match the person’s skin tone. This results in full-body textures with improved coloring in the hand regions as displayed in Figure 3-5.

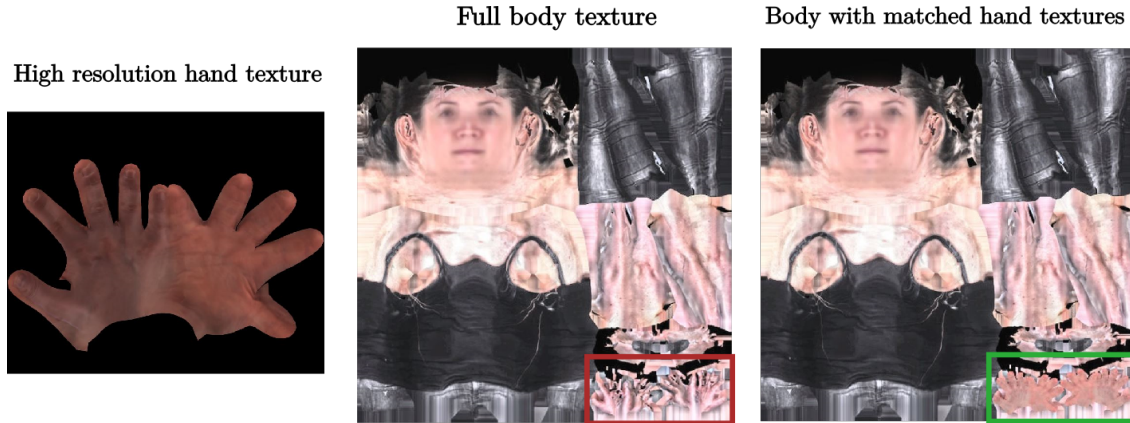


Figure 3-5: Textures from full body scans typically display missing and erroneous color values in the hand region. We use high-resolution hand scans color-aligned with the person’s face skin to inpaint the target hand region (see lower right of each body texture image, outlined in red before and in green after inpainting) as described in Section 3.2.2.

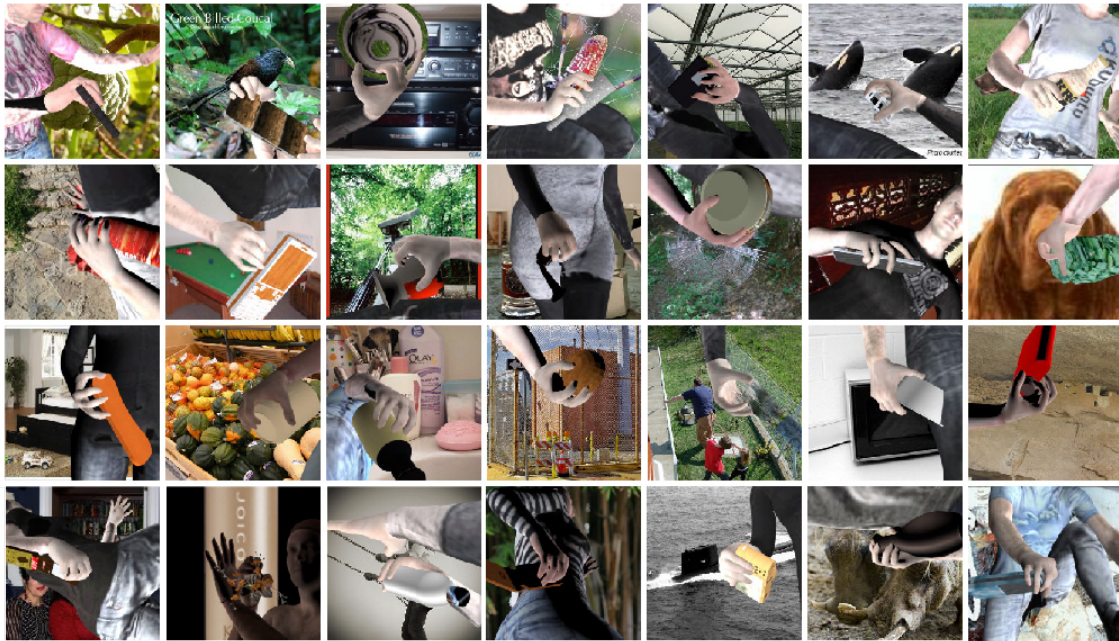


Figure 3-6: **ObMan**: large-scale synthetic dataset of hand-object interactions. We pose the MANO hand model Romero et al. (2017) to grasp a given object mesh using GraspIt Miller and Allen (2004), see Section 3.2.1. The scenes are rendered with variation in texture, lighting, and background, as described in Section 3.2.2.

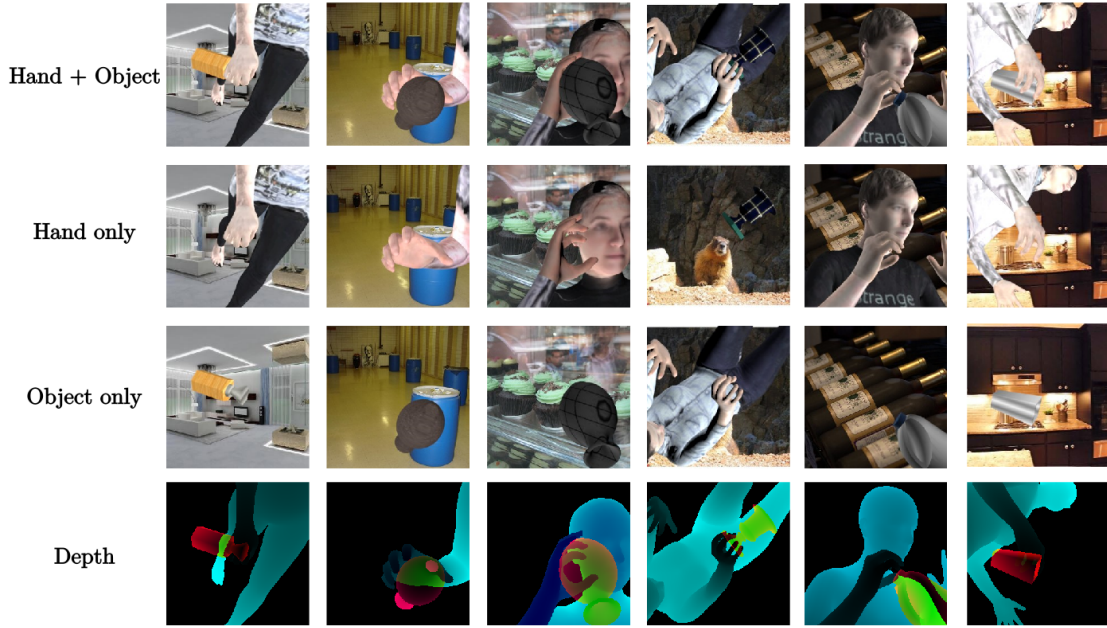


Figure 3-7: We render object-only and hand-only images for each sample in the ObMan dataset along with depth maps for each of the hand-only object-only and joint configuration, which we store in different color channels as an image along with minimum and maximum depth values for efficiency.

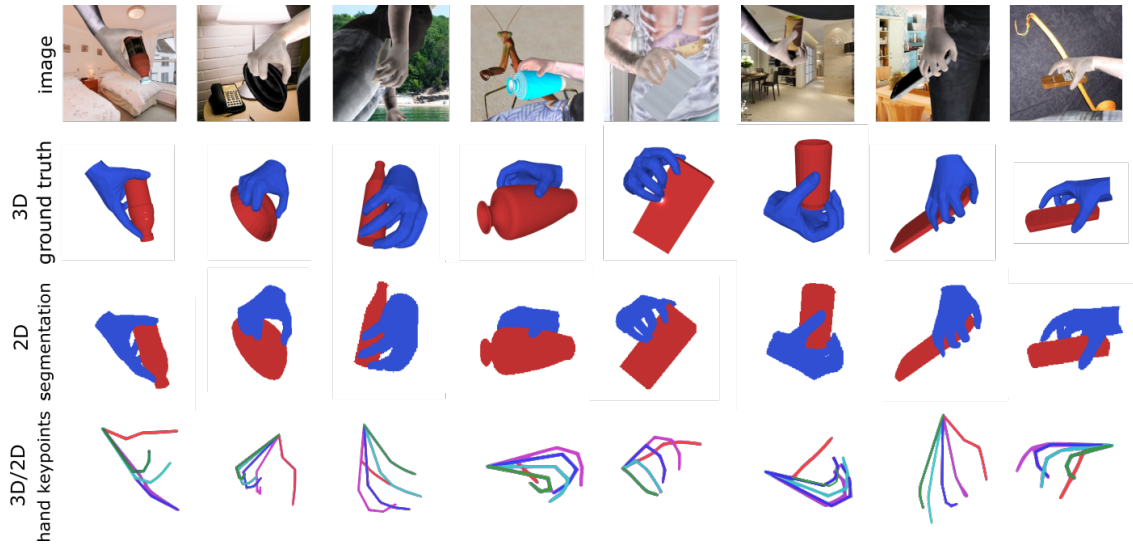


Figure 3-8: Our **ObMan** dataset provides *synthetic* images with pixel-accurate segmentation maps, 3D hand joints as well as hand and object meshes in camera coordinates.

Rendering. We render the images using Blender Blender Online Community (2018). Background images are sampled from both the LSUN Yu et al. (2015) and ImageNet Deng et al. (2009) datasets. In order to ensure the hand and objects are visible we discard configurations if less than 100 pixels of the hand or if less than 40% of the object is visible. For each hand-object configuration, we render object-only, hand-only, and hand-object images, as well as the corresponding segmentation and depth maps. Examples of the resulting outputs are presented in Figure 3-7, while Figure 3-8 illustrates automatically acquired 2D and 3D labels.

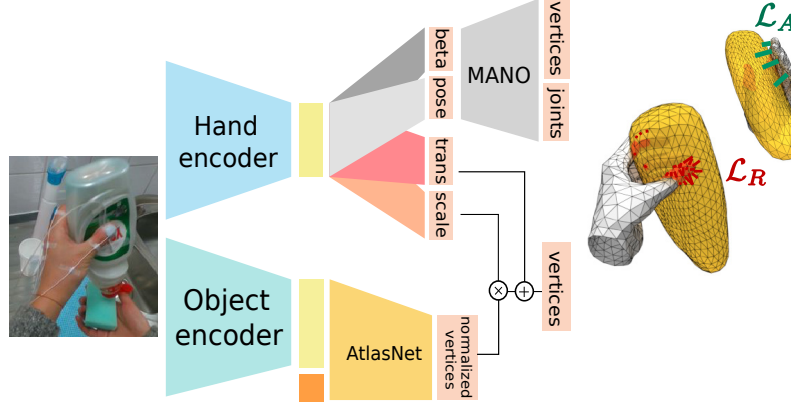


Figure 3-9: Our model predicts the hand and object meshes in a single forward pass in an end-to-end framework. The repulsion loss \mathcal{L}_R penalizes interpenetration while the attraction loss \mathcal{L}_A encourages the contact regions to be in contact with the object.

3.3 Learning grasp reconstruction

Our goal is to develop a model which can jointly estimate the object shape and hand pose from a single color image. Recent work demonstrate that CNNs with residual connections He et al. (2015) can reliably extract 3D shape information from image pixels (Groueix et al. (2018b); Kanazawa et al. (2018a); Pavlakos et al. (2018); Yang et al. (2018)). In order to seamlessly reason about contacts, we choose to reconstruct both the object and grasping hand as surface meshes. As illustrated in Figure 3-9, we design a neural network architecture that reconstructs the hand-object configuration in a single forward pass from a rough image crop of a left hand holding an object. Our network architecture is split into two branches. The first branch reconstructs the object shape in a normalized coordinate space. The second branch predicts the hand mesh as well as the information necessary to transfer the object to the hand-relative coordinate system. Each branch has a ResNet18 He et al. (2015) encoder pre-trained on ImageNet Deng et al. (2009). In the following, we detail the three components of our method: hand mesh estimation in Section 3.3.1, object mesh estimation in Section 3.3.2, and the contact between the two meshes in Section 3.3.3.

3.3.1 Regressing hand parameters

We aim to recover accurate 3D hand information from a single RGB image. In order to reason about contacts, we want to regress the dense hand surface while leveraging the strength of CNN backbones. In Section 2.2.3 we presented the advantages of parametric hand models such as MANO, which capture the shape and pose variations in a low-dimensional parameters. Regressing these parameters in a learned framework has the potential to output anatomically valid hand meshes while spanning the range of statistically plausible motion.

Following methods which integrate the SMPL parametric body model Loper et al. (2015) as a network layer Kanazawa et al. (2018a); Pavlakos et al. (2018), we integrate the MANO hand model Romero et al. (2017) as a differentiable layer. MANO is a statistical model that maps pose (θ) and shape (β) parameters to a mesh. While the pose parameters capture the angles between hand joints, the shape parameters control the person-specific deformations of the hand; as illustrated in Figure 2-7.

PCA pose parameters regression. Hand pose lives in a low-dimensional subspace Lin et al. (2000); Romero et al. (2017). Instead of predicting the full 45-dimensional pose space, we predict 30 pose PCA components of the MANO model. We found that performance saturates at 30 PCA components and keep this value for all our experiments (see Section 3.4.3).

Supervision on vertex and joint positions ($\mathcal{L}_{V_{Hand}}, \mathcal{L}_J$). The hand encoder produces an encoding Φ_{Hand} from an image. Given Φ_{Hand} , a fully connected network regresses θ and β . We integrate the mesh generation as a differentiable network layer that takes θ and β as inputs and outputs the hand vertices V_{Hand} and 16 hand joints. In addition to MANO joints, we select 5 vertices on the mesh as fingertips to obtain 21 hand keypoints J . We define the supervision on the vertex positions ($\mathcal{L}_{V_{Hand}}$) and joint positions (\mathcal{L}_J) to enable training on datasets where a ground truth hand surface is not available. Both losses are defined as the L2 distance to the ground truth. We use root-relative 3D positions as supervision for $\mathcal{L}_{V_{Hand}}$ and \mathcal{L}_J . Unless otherwise specified, we use the wrist defined by MANO as the root joint.

Regularization on hand shape (\mathcal{L}_β). Sparse supervision can cause extreme mesh deformations when the hand shape is unconstrained. We therefore use a regularizer, $\mathcal{L}_\beta = \|\beta\|^2$, on the hand shape to constrain it to be close to the average shape in the MANO training set, which corresponds to $\beta = \vec{0} \in \mathbb{R}^{10}$.

The resulting hand reconstruction loss \mathcal{L}_{Hand} is the summation of all $\mathcal{L}_{V_{Hand}}$, \mathcal{L}_J and \mathcal{L}_β terms:

$$\mathcal{L}_{Hand} = \mathcal{L}_{V_{Hand}} + \mathcal{L}_J + \mathcal{L}_\beta. \quad (3.2)$$

Our experiments indicate benefits for all three terms (see Section 3.4.3). Our hand branch also matches state-of-the-art performance on a standard benchmark for 3D hand pose estimation (see Section 3.4.3).

3.3.2 Object mesh estimation

Even across a single object category, everyday object instances come in diverse shapes, as we introduced in Section 1.2.1. Capturing instance-specific shape variations and dense surfaces is critical to accurately model contacts and useful for applications such as robotic assistance in daily tasks. In order to easily perform collision checking between the hand and the object, we choose to predict the object shape by deforming watertight mesh templates. The watertight property allows the use of efficient parallel computing for ray-triangle intersection checking (Akenine-Möller and Trumbore (1997)). In this chapter, we follow recent methods Kato et al. (2018); Wang et al. (2018) and focus on genus 0 topologies and predict object meshes by deforming a spherical mesh, automatically maintaining the watertight property of the original template. While this assumption restricts the range of objects which can be exactly modeled by our approach, we observed in early experiments that deforming a simple template led to improved generalization across categories.

Object shape estimation. We use AtlasNet Groueix et al. (2018b) as the object prediction component of our neural network architecture. AtlasNet takes as input the concatenation of point coordinates sampled either on a set of square patches or on a sphere, and image features Φ_{Obj} . It uses a fully connected network to output new coordinates on the surface of the reconstructed object. AtlasNet explores two sampling strategies: sampling points from a sphere and sampling points from a set of squares. Preliminary experiments showed better generalization to unseen classes when input points were sampled on a sphere. In all our experiments we deform an icosphere of subdivision level 3 which has 642 vertices. AtlasNet was initially designed to reconstruct meshes in a canonical view. In our model, meshes are reconstructed in view-centered coordinates. We experimentally verified that AtlasNet can accurately reconstruct meshes in this setting (see Section 3.4.4). Following AtlasNet, the supervision for object vertices is defined by the symmetric Chamfer loss between the predicted vertices and points randomly sampled on the ground truth external surface of the object.

Regularization on object shape ($\mathcal{L}_E, \mathcal{L}_L$). In order to reason about the inside and outside of the object, it is important to predict meshes with well-defined surfaces and good quality triangulations. However AtlasNet does not explicitly enforce constraints on mesh quality. We find that when learning to model a limited number of object shapes, the triangulation quality is preserved. However, when training on the larger variety of objects of ObMan, we find additional regularization on the object meshes beneficial. Following Groueix et al. (2018a); Kanazawa et al. (2018b); Wang et al. (2018) we employ two losses that penalize irregular meshes. We penalize edges with lengths different from the average edge length

with an edge-regularization loss, \mathcal{L}_E . We further introduce a curvature-regularizing loss, \mathcal{L}_L , based on Kanazawa et al. (2018b), which encourages the curvature of the predicted mesh to be similar to the curvature of a sphere.

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.

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:

$$\mathcal{L}_E = \frac{1}{|\mathcal{E}_L|} \sum_{l \in \mathcal{E}_L} |l^2 - \mu(\mathcal{E}_L^2)|, \quad (3.3)$$

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. Note that this loss is equal to zero when all edges have the same length and positive otherwise.

Combination of reconstruction and regularization losses. We balance the weights of \mathcal{L}_E and \mathcal{L}_L by weights μ_E and μ_L respectively, which we empirically set to 2 and 0.1. These two losses together improve the quality of the predicted meshes, as we show in Figure 3-14. Additionally, when training on the ObMan dataset, we first train the network to predict normalized objects, and then freeze the object encoder and the AtlasNet decoder while training the hand-relative part of the network. When training the objects in normalized coordinates, noted with n , the total object loss is:

$$\mathcal{L}_{Object}^n = \mathcal{L}_{V_{Obj}}^n + \mu_L \mathcal{L}_L + \mu_E \mathcal{L}_E. \quad (3.4)$$

Hand-relative coordinate system ($\mathcal{L}_S, \mathcal{L}_T$). Following AtlasNet Groueix et al. (2018b), we first predict the object in a normalized scale by offsetting and scaling the ground truth vertices so that the object is inscribed in a sphere of fixed radius. However, as we focus on hand-object interactions, we need to estimate the object position and scale relative to the hand. We therefore predict translation and scale in two branches, which output the three offset coordinates for the translation (i.e., x, y, z) and a scalar for the object scale. We define $\mathcal{L}_T = \|T - \hat{T}\|_2^2$ and $\mathcal{L}_S = \|S - \hat{S}\|_2^2$, where \hat{T} and \hat{S} are the predicted translation and scale. T is the ground truth object centroid in hand-relative coordinates and S is the

ground truth maximum radius of the centroid-centered object.

Supervision on object vertex positions ($\mathcal{L}_{V_{Obj}}^n, \mathcal{L}_{V_{Obj}}$). We multiply the AtlasNet decoded vertices by the predicted scale and offset them according to the predicted translation to obtain the final object reconstruction. Chamfer loss ($\mathcal{L}_{V_{Obj}}$) is applied after translation and scale are applied. When training in hand-relative coordinates the loss becomes:

$$\mathcal{L}_{Object} = \mathcal{L}_T + \mathcal{L}_S + \mathcal{L}_{V_{Obj}}. \quad (3.5)$$

3.3.3 Contact loss

So far, the prediction of hands and objects does not leverage the constraints that guide objects interacting in the physical world. Specifically, it does not account for our prior knowledge that objects can not interpenetrate each other and that, when grasping objects, contacts occur at the surface between the object and the hand. We formulate these contact constraints as a differentiable loss, $\mathcal{L}_{Contact}$, which can be directly used in the end-to-end learning framework. We incorporate this additional loss using a weight parameter μ_C , which we set empirically to 10.

We rely on the following definition of distances between points. $d(v, V_{Obj}) = \inf_{w \in V_{Obj}} \|v - w\|_2$ denotes distances from point to set and $d(C, V_{Obj}) = \inf_{v \in C} d(v, V_{Obj})$ denotes distances from set to set. Moreover, we define a common penalization function $l_\alpha(x) = \alpha \tanh\left(\frac{x}{\alpha}\right)$, where α is a characteristic distance of action.

Repulsion (\mathcal{L}_R). We define a repulsion loss (\mathcal{L}_R) that penalizes hand and object *interpenetration*. To detect interpenetration, we first detect hand vertices that are inside the object. Since the object is a deformed sphere, it is watertight. We therefore cast a ray from the hand vertex and count the number of times it intersects the object mesh to determine whether it is inside or outside the predicted mesh Akenine-Möller and Trumbore (1997). \mathcal{L}_R affects all hand vertices that belong to the interior of the object, which we denote $\text{Int}(Obj)$. The repulsion loss is defined as:

$$\mathcal{L}_R(V_{Obj}, V_{Hand}) = \sum_{v \in V_{Hand}} \mathbb{1}_{v \in \text{Int}(V_{Obj})} l_r(d(v, V_{Obj})),$$

where r is the repulsion characteristic distance, which we empirically set to 2cm in all experiments.

Attraction (\mathcal{L}_A). We further define an attraction loss (\mathcal{L}_A) to penalize cases in which hand vertices are in the vicinity of the object but the surfaces are *not* in contact. This loss

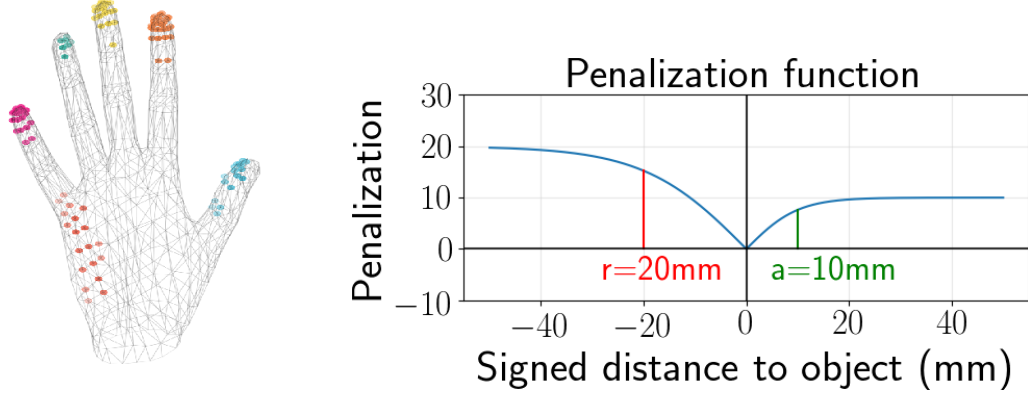


Figure 3-10: Left: Estimated contact regions from ObMan. We find that points that are often involved in contacts can be clustered into 6 regions on the palmar surface of the hand. Right: Generic shape of the penalization function emphasizing the role of the characteristic distances.

is applied only to vertices which belong to the exterior of the object $\text{Ext}(\text{Obj})$.

We compute statistics on the automatically-generated grasps described in the next section to determine which vertices on the hand are frequently involved in contacts. We compute for each MANO vertex how often across the dataset it is in the immediate vicinity of the object (defined as less than 3mm away from the object’s surface). We find that by identifying the vertices that are close to the objects in at least 8% of the grasps, we obtain 6 regions of connected vertices $\{C_i\}_{i \in [1,6]}$ on the hand which match the 5 fingertips and part of the palm of the hand, as illustrated in Figure 3-10 (left). The attraction term \mathcal{L}_A penalizes distances from each of the regions to the object, allowing for sparse guidance towards the object’s surface:

$$\mathcal{L}_A(V_{\text{Obj}}, V_{\text{Hand}}) = \sum_{i=1}^6 l_a(d(C_i \cap \text{Ext}(\text{Obj}), V_{\text{Obj}})). \quad (3.6)$$

We set a to 1cm in all experiments. For regions that are further from the hand than a threshold a , the attraction will significantly decrease and become negligible as the distance to the object further increases, see Figure 3-10 (right).

Our final contact loss $\mathcal{L}_{\text{Contact}}$ is a weighted sum of the attraction \mathcal{L}_A and the repulsion \mathcal{L}_R terms:

$$\mathcal{L}_{\text{Contact}} = \lambda_R \mathcal{L}_R + (1 - \lambda_R) \mathcal{L}_A, \quad (3.7)$$

where $\lambda_R \in [0, 1]$ is the contact weighting coefficient, e.g., $\lambda_R = 1$ means only the repulsion term is active. We show in our experiments that the balancing between attraction and repulsion is very important for physical quality.

Our network is first trained with $\mathcal{L}_{\text{Hand}} + \mathcal{L}_{\text{Object}}$. We then continue training with $\mathcal{L}_{\text{Hand}} + \mathcal{L}_{\text{Object}} + \mu_C \mathcal{L}_{\text{Contact}}$ to improve the physical quality of the hand-object interaction.

Subsection 3.3.4 gives further implementation details.

3.3.4 Implementation details

Model optimization. For all our experiments, we use the Adam optimizer Kingma and Ba (2014). 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 Deng et al. (2009) pre-trained ResNet18 He et al. (2015).

Training procedure. For the final model trained on ObMan, we first train the (normalized) object branch using \mathcal{L}_{Object}^n 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 3.3.2. 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 FPHAB_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 Kingma and Ba (2014) to zero, as we find that momentum affects negatively the training stability when we include the contact loss.

Weight balancing. In all experiments, we keep the relative weights between different losses as detailed in Section 3.3.2 and normalize them so that the sum of all the weights equals 1.

Runtime. At test time, our model can process 20 fps on a Titan X GPU.

3.4 Experiments

We first define the evaluation metrics and the datasets (Sections 3.4.1, 3.4.2) for our experiments. We present preliminary analysis for hand-only reconstruction in Section 3.4.3 and object-only reconstruction in Section 3.4.4. We then analyze the effects of occlusions (Section 3.4.5) and the contact loss (Section 3.4.6). Finally, we present our transfer learning experiments from synthetic to real domain (Sections 3.4.7, 3.4.8).

3.4.1 Evaluation metrics

Our output is structured, and a single metric does not fully capture performance. We therefore rely on multiple evaluation metrics.

Hand error. For hand reconstruction, we compute the mean end-point error (mm) over 21 joints following Zimmermann and Brox (2017).

Object error. Following AtlasNet Groueix et al. (2018b), we measure the accuracy of object reconstruction by computing the symmetric Chamfer distance (mm) between points sampled on the ground truth mesh and vertices of the predicted mesh.

Contact. To measure the physical quality of our joint reconstruction, we use the following metrics.

Penetration depth (mm), Intersection volume (cm^3): Hands and objects should not share the same physical space. To measure whether this rule is violated, we report the intersection volume between the object and the hand as well as the penetration depth. To measure the intersection volume of the hand and object we voxelize the hand and object using a voxel size of 0.5cm. If the hand and the object collide, the penetration depth is the maximum of the distances from hand mesh vertices to the object’s surface. In the absence of collision, the penetration depth is 0.

Simulation displacement (mm): Following Tzionas et al. (2016), we use physics simulation to evaluate the quality of the produced grasps. This metric measures the average displacement of the object’s center of mass in a simulated environment Coumans (2013) assuming the hand is fixed and the object is subjected to gravity. Details on the setup and the parameters used for the simulation can be found in Tzionas et al. (2016). Good grasps should be stable in simulation. However, stable simulated grasps can also occur if the forces resulting from the collisions balance each other. For estimating grasp quality, simulated displacement must be analyzed in conjunction with a measure of collision. If both displacement in simulation and penetration depth are decreasing, there is strong evidence that the physical quality of the grasp is improving (see Section 3.4.6 for an analysis). The reported metrics are averaged across the dataset.

3.4.2 Datasets

We present the datasets we use to evaluate our models. Statistics for each dataset are summarized in Table 3.1.

First-person hand benchmark (FPHAB). This dataset Garcia-Hernando et al. (2018) is a recent video collection providing 3D hand annotations for a wide range of hand-object interactions. The joints are automatically annotated using magnetic sensors strapped on

	ObMan	FPHAB	FPHAB _C	HIC
#frames	141K/6K	8420/9103	5077/5657	251/307
#video sequences	-	115/127	76/88	2/2
#object instances	1947/411	4	3	2
real	no	yes	yes	yes

Table 3.1: Dataset details for train/test splits.

the hands, and which are visible on the RGB images. 3D mesh annotations are provided for four objects: three different bottles and a salt box. In order to ensure that the object being interacted with is unambiguously defined, we filter frames in which the manipulating hand is further than 1cm away from the manipulated object. We refer to this filtered dataset as FPHAB. As the milk bottle is a genus-1 object and is often grasped by its handle, we exclude this object from the experiments we conduct on contacts. We call this subset FPHAB_C. We use the same subject split as Garcia-Hernando et al. (2018), therefore, each object is present in both the training and test splits.

The object annotations for this dataset suffer from some imprecisions. To investigate the range of the object ground truth error, we measure the penetration depth of the hand skeleton in the object for each hand-object configuration. We find that on the training split of FPHAB, the average penetration depth is 11.0mm (std=8.9mm). While we still report quantitative results on objects for completeness, the ground truth errors prevent us from drawing strong conclusions from reconstruction metric fluctuations on this dataset.

Hands in action dataset (HIC). We use a subset of the HIC dataset Tzionas et al. (2016) which has sequences of a single hand interacting with objects. This gives us 4 sequences featuring manipulation of a sphere and a cube. We select the frames in which the hand is less than 5mm away from the object. We split this dataset into 2 training and 2 test sequences with each object appearing in both splits and restrict our predictions to the frames in which the minimal distance between hand and object vertices is below 5mm. For this dataset the hand and object meshes are provided. We fit MANO to the provided hand mesh, allowing for dense point supervision on both hands and objects.

CORe50. CORe50 Lomonaco and Maltoni (2017) is a dataset which contains hand-object interactions with an emphasis on the variability of objects and backgrounds. However no 3D hand or object annotation is available. We therefore present qualitative results on this dataset.

	ObMan	FPHAB	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 3.2: 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}_β). We show the results of training and testing on our synthetic ObMan dataset, as well as the real datasets FPHAB Garcia-Hernando et al. (2018) and StereoHands Zhang et al. (2016).

3.4.3 Hand pose estimation

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

Loss study on MANO As explained in Section 3.3.1, 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}_β . The shape is only predicted in the presence of \mathcal{L}_β . 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 3.2 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 FPHAB Garcia-Hernando et al. (2018) and StereoHands Zhang et al. (2016). We find that predicting β while regularizing it with \mathcal{L}_β 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}_β when only keypoint supervision is provided.

MANO pose representation As described in Section 3.3.1, 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 3.3. 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

#PCA comps.	6	15	30	45
FPHAB	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 3.3: 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.

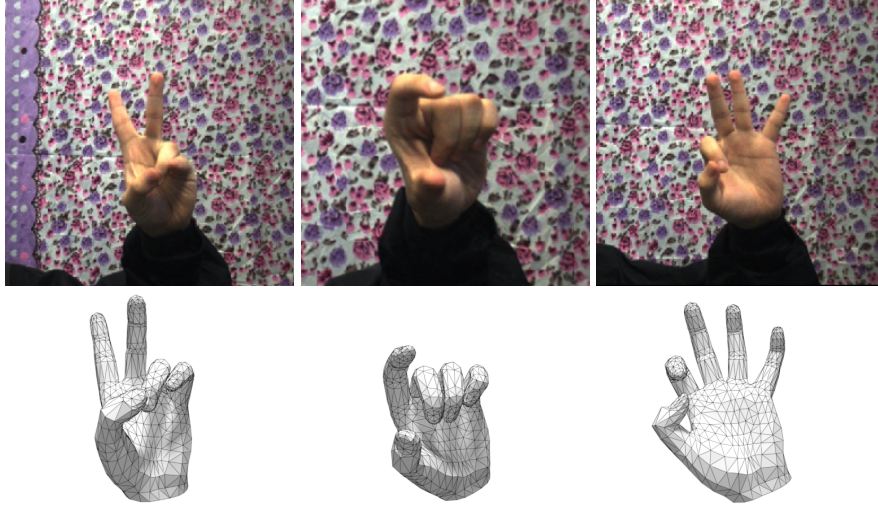


Figure 3-11: Qualitative results on the test sequence of the StereoHands dataset.

for all other experiments in this chapter.

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 Zhang et al. (2016), and follow the evaluation protocol of Iqbal et al. (2018); Mueller et al. (2018); Zimmermann and Brox (2017) 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 3-12 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 Cai

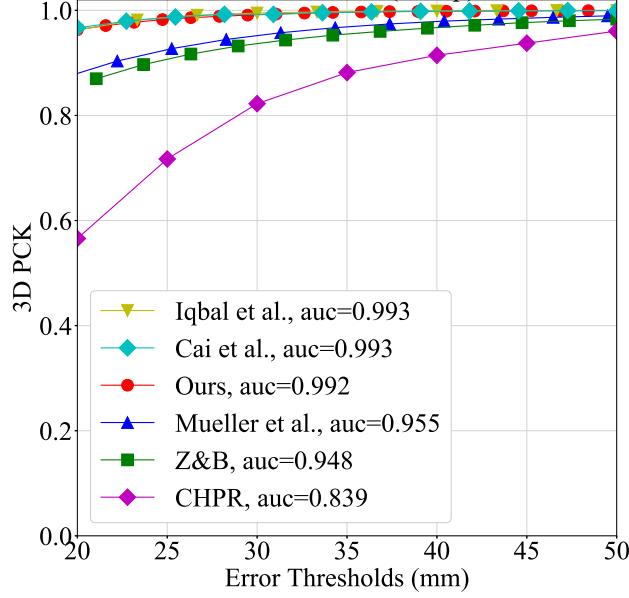


Figure 3-12: We compare our root-relative 3D hand pose estimation on Stereohands to the state-of-the-art methods from Iqbal et al. (2018), Cai et al. (2018), Mueller et al. (2018), Zimmermann and Brox (2017), and CHPR Sun et al. (2015).

et al. (2018); Iqbal et al. (2018); Mueller et al. (2018); Sun et al. (2015); Zimmermann and Brox (2017) that only predicts sparse skeleton keypoints, our model produces a *dense* hand mesh. Figure 3-11 presents some qualitative results from this dataset.

3.4.4 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 and (ii) with regularization losses.

Canonical versus camera view reconstruction. As explained in Section 3.3.2, we perform object reconstructions in the camera coordinate frame. To validate that AtlasNet Groueix et al. (2018b) can successfully predict objects in camera view as well as in canonical view, we reproduce the training setting of the original paper Groueix et al. (2018b). We use the setting where 2500 points are sampled on a sphere and train on the rendered images from ShapeNet Choy et al. (2016). 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 Kingma and Ba (2014) 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

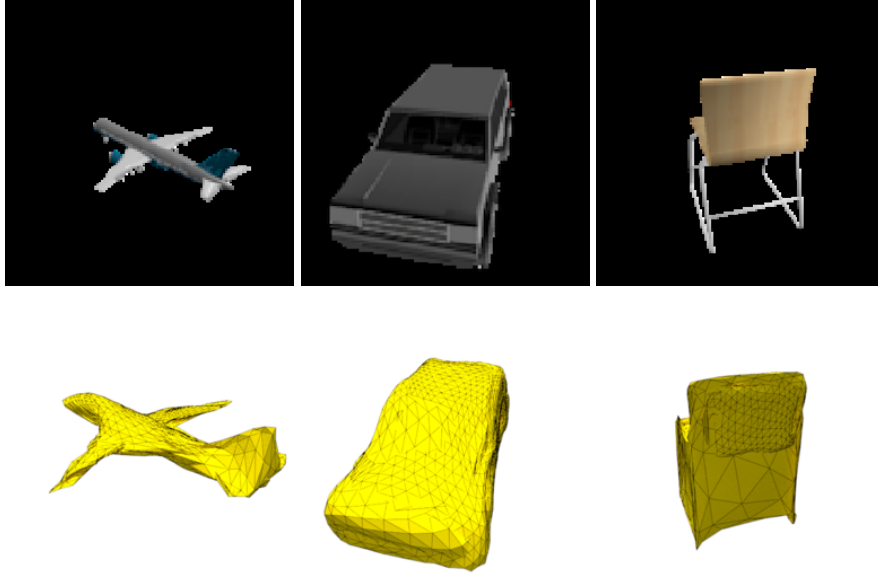


Figure 3-13: Renderings from ShapeNet models and our corresponding reconstructions in camera view.

	Object error
Canonical view Groueix et al. (2018b)	4.87
Canonical view (ours)	4.88
Camera view (ours)	4.88

Table 3.4: Chamfer loss ($\times 1000$) 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 Groueix et al. (2018b) on their code page <https://github.com/ThibaultGROUEIX/AtlasNet>.

$\mathcal{L}_{V_{Obj}} = \frac{1}{2}(\sum_p \min_q \|p - q\|_2^2 + \sum_q \min_p \|q - p\|_2^2)$ 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 3-13.

For better readability they also multiply the Chamfer loss by 1000. In order to provide results directly comparable with the original paper Groueix et al. (2018b), we also report numbers with the same scaling in Table 3.4. Table 3.4 reports the Chamfer distances for their released model, our reimplementation in canonical 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.

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 3-13. Following recent work on mesh estimation from image inputs Groueix et al. (2018a); Kanazawa et al. (2018b); Wang et al. (2018), we introduce regularization terms on the object mesh.

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 3-14 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.

3.4.5 Effect of occlusions

Experimental setup for occlusion study. We study the effect of objects occluding hands by training two networks, one trained on hand-only images and one on hand-object images. For each sample in our synthetic dataset, in addition to the hand-object image (HO-img) we render two images of the corresponding isolated and unoccluded hand (H-img) or object (O-img). With this setup, we can systematically study the effect of occlusions on ObMan, which would be impractical outside of a synthetic setup. Note that reproducing such a setup would be impractical outside of a synthetic environment.

We report performance on both unoccluded and occluded images. A symmetric setup is applied to study the effect of hand occlusions on objects by training two additional networks on object-only and hand-object images. Since the hand-relative coordinates are not applicable to experiments with object-only images, we study the normalized shape reconstruction, centered on the object centroid, and scaled to be inscribed in a sphere of radius 1.

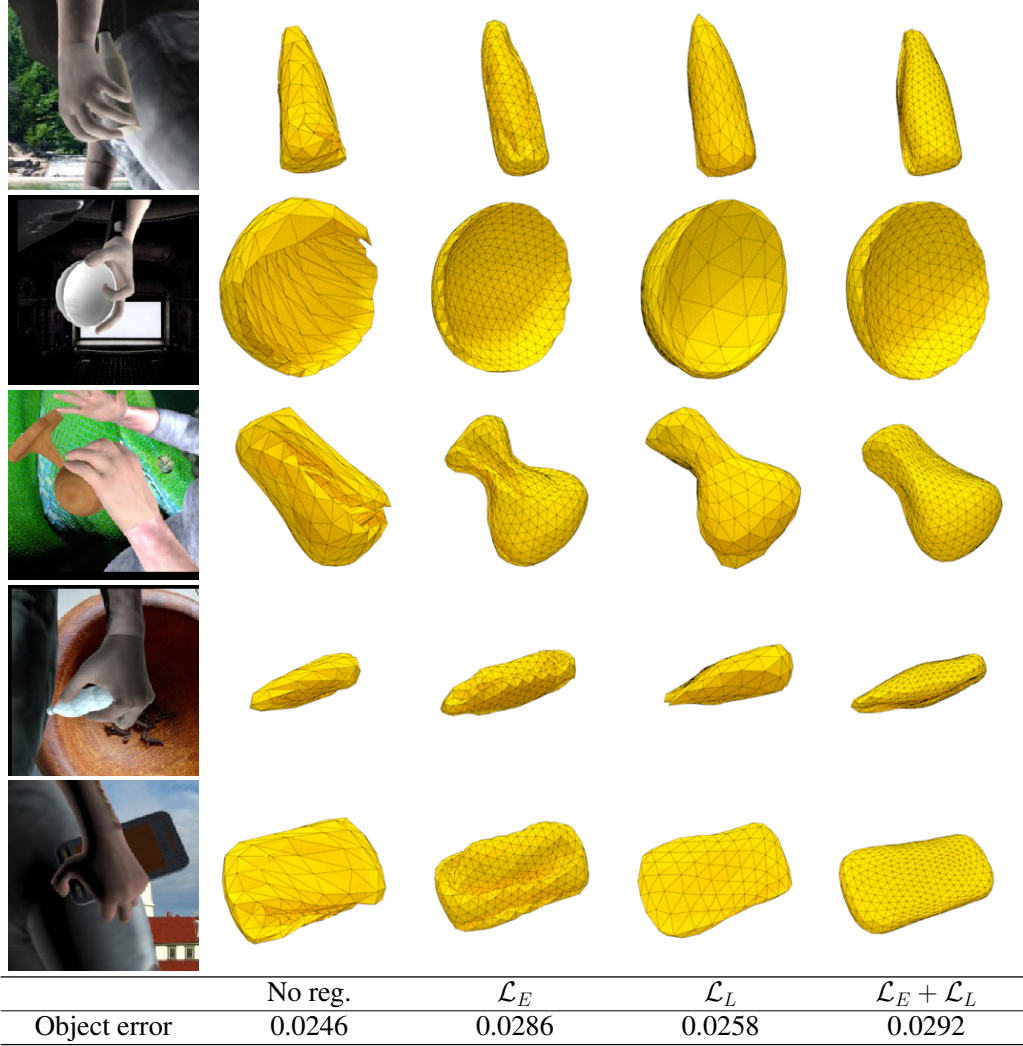


Figure 3-14: We show the benefits from each term of the regularization. Using both the \mathcal{L}_E and \mathcal{L}_L in conjunction improves the visual quality of the predicted triangulation while preserving the shape of the object.

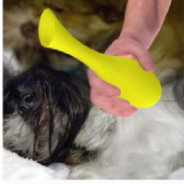
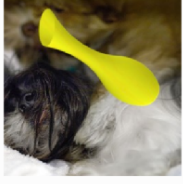

HO-image			O-image			H-image		
								
Training	Evaluation images		Training	Evaluation images		Training	Evaluation images	
	H-img	HO-img		O-img	HO-img		O-img	HO-img
H-img (\mathcal{L}_H)	10.3	14.1	O-img (\mathcal{L}_O)	0.0242	0.0722			
HO-img (\mathcal{L}_H)	11.7	11.6	HO-img (\mathcal{L}_O)	0.0319	0.0302			

Table 3.5: We first show that training with occlusions is important when targeting images of hand-object interactions.

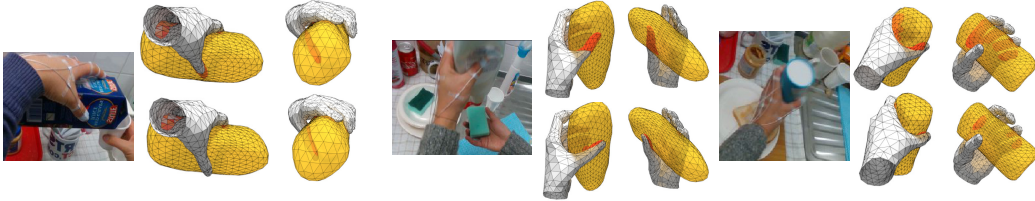


Figure 3-15: Qualitative comparison between *with* (bottom) and *without* (top) contact on FPHAB_C. Note the improved contact and reduced penetration, highlighted with red regions, with our contact loss.

Effect of occlusions on hand and object reconstruction accuracy. Unsurprisingly, the best performance is obtained when both training and testing on unoccluded images as shown in Table 3.5. When both training and testing on occluded images, reconstruction errors for hands and objects drop significantly, by 12% and 25% respectively. This validates the intuition that estimating hand pose and object shape in the presence of occlusions is a harder task. We observe that for both hands and objects, the most challenging setting is training on unoccluded images while testing on images with occlusions. This shows that training with occlusions is crucial for accurate reconstruction of hands-object configurations.

3.4.6 Effect of contact loss

In the absence of explicit physical constraints, the predicted hands and objects have an average penetration depth of 9mm for ObMan and 19mm for FPHAB_C (see Table 3.6). The presence of interpenetration at test time shows that the model is not implicitly learning the physical rules governing hand-object manipulation. The differences in physical metrics between the two datasets can be attributed to the higher reconstruction accuracy for ObMan but also to the noisy object ground truth in FPHAB_C which produces penetrated and likely unstable ‘ground truth’ grasps.

	ObMan Dataset				
	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume
No contact loss	11.6	641.5	9.5	31.3	12.3
Only attraction ($\lambda_R = 0$)	11.9	637.8	11.8	26.8	17.4
Only repulsion ($\lambda_R = 1$)	12.0	639.0	6.4	38.1	8.1
Attraction + Repulsion ($\lambda_R = 0.5$)	11.6	637.9	9.2	30.9	12.2

	FPHAB _C Dataset				
	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume
No contact loss	28.1 \pm 0.5	1579.2 \pm 66.2	18.7 \pm 0.6	51.2 \pm 1.7	26.9 \pm 0.2
Only attraction ($\lambda_R = 0$)	28.4 \pm 0.6	1586.9 \pm 58.3	22.7 \pm 0.7	48.5 \pm 3.2	41.2 \pm 0.3
Only repulsion ($\lambda_R = 1$)	28.6 \pm 0.8	1603.7 \pm 49.9	6.0 \pm 0.3	53.9 \pm 2.3	7.1 \pm 0.1
Attraction + Repulsion ($\lambda_R = 0.5$)	28.8 \pm 0.8	1565.0 \pm 65.9	12.1 \pm 0.7	47.7 \pm 2.5	17.6 \pm 0.2

Table 3.6: We experiment with each term of the contact loss. Attraction (\mathcal{L}_A) encourages contacts between close points while repulsion (\mathcal{L}_R) penalizes interpenetration. λ_R is the repulsion weight, balancing the contribution of the two terms.

Fine-tuning with differentiable contact losses. In Figure 3-16, we study the effect of introducing our contact loss as a fine-tuning step. We linearly interpolate λ_R in $\llbracket 0, 1 \rrbracket$ to explore various relative weightings of the attraction and repulsion terms. We find that using \mathcal{L}_R in isolation efficiently minimizes the maximum penetration depth, reducing it by 33% for ObMan and 68% for FPHAB_C. This decrease occurs at the expense of the stability of the grasp in simulation. Symmetrically, \mathcal{L}_A stabilizes the grasps in simulation, but produces more collisions between hands and objects. We find that equal weighting of both terms ($\mathcal{L}_R = 0.5$) improves *both* physical measures without negatively affecting the reconstruction metrics on both the synthetic and the real datasets, as is shown in Table 3.6 (last row). We observe different results in terms of simulation displacements across runs on the relatively small FPHAB_C dataset. For each metric we report the means and standard deviations for 10 random seeds.

We find that on the synthetic dataset, decreased penetration is systematically traded for simulation instability whereas for FPHAB_C increasing λ_R from 0 to 0.5 decreases depth penetration *without* affecting the simulation stability. Furthermore, for $\lambda_R = 0.5$, we observe significant qualitative improvements on FPHAB_C as seen in Figure 3-15.

3.4.7 Synthetic to real transfer

Advantage of pre-training on synthetic data. Large-scale synthetic data can be used to pre-train models in the absence of suitable real datasets. We investigate the advantages of pre-training on ObMan when targeting FPHAB and HIC. We investigate the effect of scarcity of real data on FPHAB by comparing pairs of networks trained using subsets of

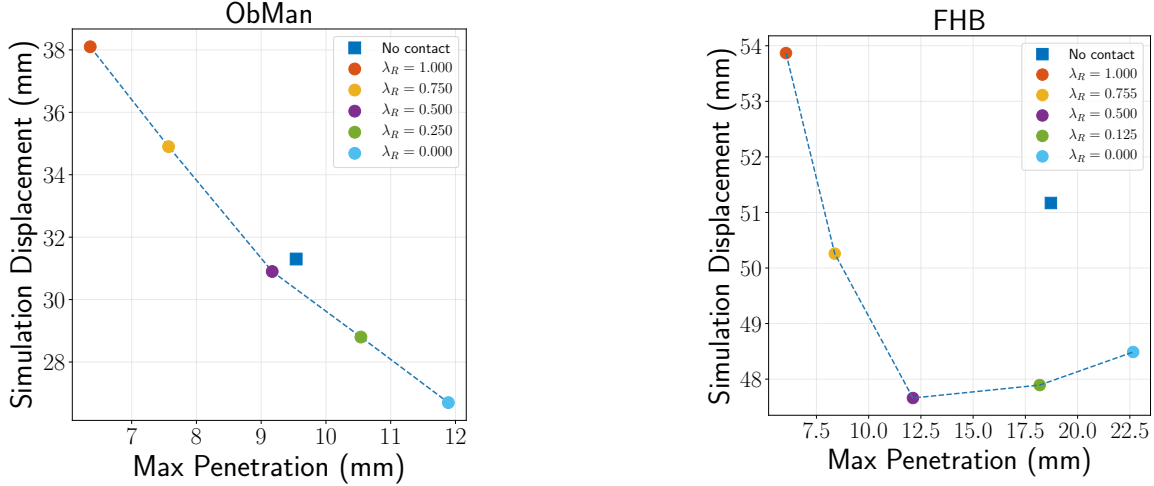


Figure 3-16: We examine the relative importance between the contact terms on the grasp quality metrics. Introducing a well-balanced contact loss improves upon the baseline on both max penetration and simulation displacement.

the real dataset. One is pre-trained on ObMan while the other is initialized randomly, with the exception of the encoders, which are pre-trained on ImageNet Deng et al. (2009). For these experiments, we do not add the contact loss and report means and standard deviations for 5 distinct random seeds. We find that pre-training on ObMan is beneficial in low data regimes, especially when less than 1000 images from the real dataset are used for fine-tuning, see Figure 3-17.

Domain gap study. The HIC training set consists of only 250 images. We experiment with pre-training on variants of our synthetic dataset. In addition to ObMan, to which we refer as (a) in Figure 3-18, we render 20K images for two additional synthetic datasets, (b) and (c), which leverage information from the training split of HIC (d). We create (b) using our grasping tool to generate automatic grasps for each of the object models of HIC and (c) using the object and pose distributions from the training split of HIC. This allows to study the importance of sampling hand-object poses from the target distribution of the real data. We explore training on (a), (b), (c) with and without fine-tuning on HIC. We find that pre-training on all three datasets is beneficial for hand and object reconstructions. The best performance is obtained when pre-training on (c). In that setup, object performance outperforms training only on real images even *before* fine-tuning, and significantly improves upon the baseline after. Hand pose error saturates after the pre-training step, leaving no room for improvement using the real data. These results show that when training on synthetic data, similarity to the target real hand and pose distribution is critical.

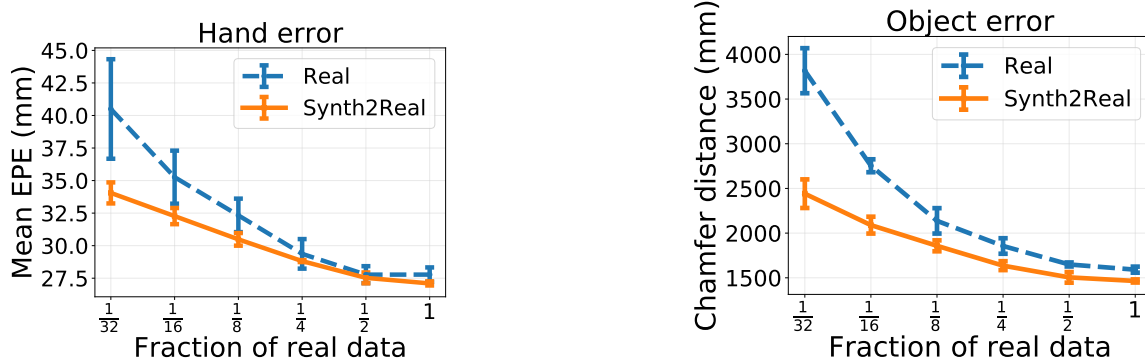


Figure 3-17: We compare training on FPHAB only (Real) and pre-training on synthetic, followed by fine-tuning on FPHAB (Synth2Real). As the amount of real data decreases, the benefit of pre-training increases. For both the object and the hand reconstruction, synthetic pre-training is critical in low-data regimes.

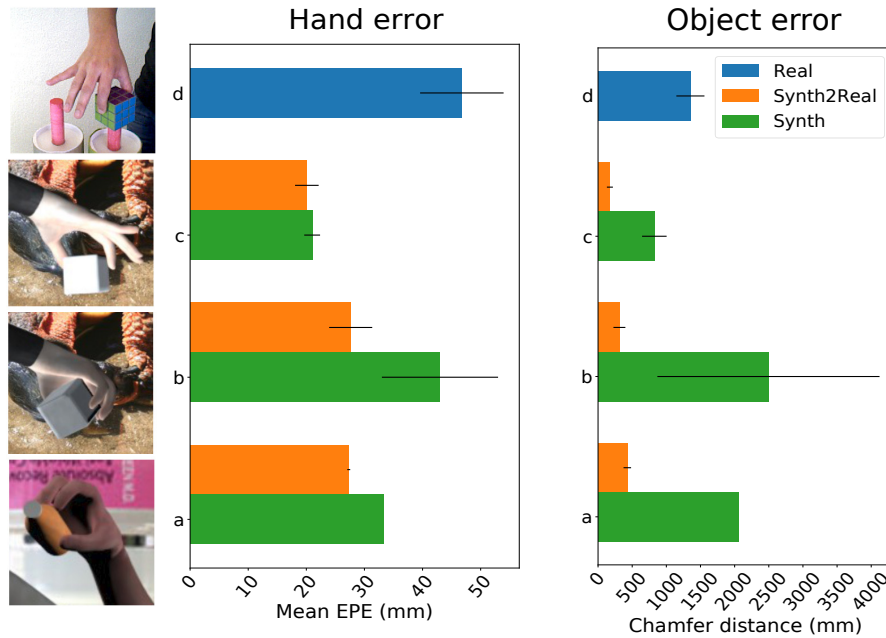


Figure 3-18: We compare the effect of training with and without fine-tuning on variants of our synthetic dataset on HIC. We illustrate each dataset (a, b, c, d) with an image sample, see text for definitions. Synthetic pre-training, whether or not the target distribution is matched, is always beneficial.

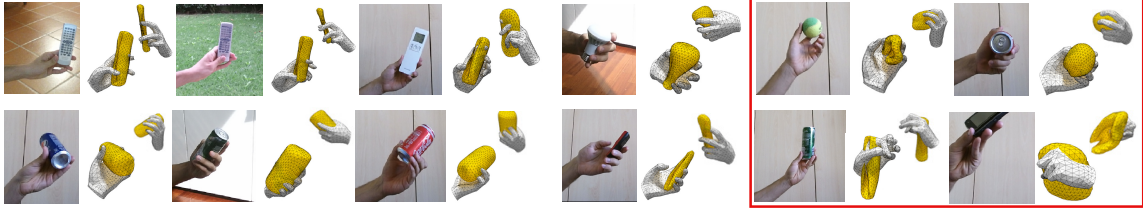


Figure 3-19: Qualitative results on CORE50. Our model, trained only on synthetic data, shows robustness to various hand poses, objects and scenes. Global hand pose and object outline are well estimated while fine details are missed. We present failure cases in the red box. Note that this model is trained on synthetic ObMan images only.



Figure 3-20: Selected 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. Each image shows manipulation of a different object model.

3.4.8 Qualitative results on CORE50

In this section, we verify the ability of our model trained on ObMan to generalize to real data *without* fine-tuning. FPHAB is a dataset with visible magnetic sensors and a specific viewpoint given the first-person perspective.

Generalization to real data. We observe empirically that our method trained only on our synthetic ObMan dataset generalizes poorly to images from this dataset. Figure 3-19 shows that our model generalizes to the CORE50 dataset Lomonaco and Maltoni (2017) across different object categories, including *light-bulb*, which does not belong to the categories our model was trained on. The global outline is well recovered in the camera view while larger mistakes occur in the perpendicular direction. We present additional qualitative results on the CORE50 Lomonaco and Maltoni (2017) dataset. We present a variety of diverse input images from CORE50 in Figure 3-20 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



Figure 3-21: To show the typical performance of our model on the CORE50 dataset Lomonaco and Maltoni (2017), We display the outputs of our method on 25 **randomly sampled** frames from this dataset. Note that the images are randomly drawn from the *subset* of CORE50 which we annotated with hand side and hand-object region of interest.

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 3-20, 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 a low-dimensional space, which implicitly regularizes hand poses, anatomical validity is not guaranteed.

To show the typical performance of our model, we present in Figure 3-21 the output of our method on randomly sampled frames from the CORE50 dataset Lomonaco and Maltoni (2017). We randomly draw input images from the *subset* of frames which we annotated with hand side and hand-object region of interest, annotations which were performed independently of model evaluation to avoid direct biases. This annotation focused on capturing a large number of different object shapes at the expense of diverse backgrounds. The qualitative examples in Figure 3-21 further illustrate the success and failure modes detailed in Figure 3-20. Namely the bias towards power grasps, reasonable approximation of various

object outlines and global hand rotation.

3.5 Conclusions

We presented an end-to-end approach for joint reconstruction of hands and objects given a single RGB image as input. Our novel contact loss enforces physical constraints on the interaction between the two meshes during training, resulting in qualitatively improved grasps during inference. Most importantly, we proposed to use automatic grasping software and computer graphics rendering to generate synthetic grasp data, and demonstrated that such images can be used to train methods to compute 3D information from RGB images.

Our proposed method for automatic generation of hand-object grasp has the advantage of being fully automatic. For any object model of graspable size, we can generate a variety of grasps which can each be rendered using a variety of viewpoints, textures, backgrounds and lighting conditions.

However, generation method produces a domain gap both in the 3D and pixel domains. In 3D, the grasps are generated and sorted using grasp quality measures Ferrari and Canny (1992) which are imperfect proxies for statistically plausible grasps. In pixel space, the limited photo-realism in textures and rendering as well as the simple composition with an image background results in images which are distinctly perceived as synthetic by any human observer.

This gap results in several observable limitations. We observed that our model trained on ObMan transfers several grasp biases from the GraspIt! grasper to the trained model. For instance, we observe that our method works best when the background and foreground are easily separable, and that precision grasps tend to be reconstructed as power grasps even in favorable viewing conditions. Several approaches can reduce the empirically observed domain gap. In the pixel domain, improved photo-realism and targeted data-augmentation can improve the generalization from real to synthetic images. In 3D, using improved automatic grasp generators could reduce the bias towards analytically valid but statistically implausible grasps.

Our synthetic data is valuable for pre-training and direct generalization in favorable conditions. However, when available, real data with manual annotations remains a useful resource. Next, in Chapter 4, we explore how to efficiently leverage manual 3D annotations when a sparse set of labelled frames is available in videos.

Chapter 4

Learning from sparse annotations

While rendering synthetic data has the advantage of being automatically scalable, application scenarios such as the ones we presented in Section 1.1 require models with high performance on *real* videos and images. When targeting generalization to real data, the domain gap has to be explicitly accounted for. Existing synthetic datasets depicting hands, such as RHD Zimmermann and Brox (2017), SynthHands Mueller et al. (2017) or ObMan cannot yet reach the fidelity and realism to generalize to real datasets. Manual annotation and optimization-based techniques for data annotation can be slow and error-prone. Due to the challenges associated with data collection, existing datasets are either real, limited in size and confined to constrained environments or synthetic and lack realism. Models trained on such data are prone to overfitting and lack generalization capabilities. Our method aims at tackling this challenge by reducing the stringent reliance on 3D annotations. To this end, in this chapter we propose a novel weakly supervised approach to joint 3D hand-object reconstruction. Our model jointly estimates the hand and object pose and reconstructs their shape in 3D, given training videos with annotations in only sparse frames on a small fraction of the dataset. Our method models the temporal nature of 3D hand and object interactions and leverages motion as a self-supervisory signal for 3D dense hand-object reconstruction.

Our contributions can be summarized as follows. (i) We present a new method for joint dense reconstruction of hands and objects in 3D. Our method operates on color images and efficiently regresses model-based shape and pose parameters in a single feed-forward pass through a neural network. (ii) We introduce a novel photometric loss that relies on the estimated optical flow between pairs of adjacent images. Our scheme leverages optical flow to warp one frame to the next, directly within the network, and exploits the visual consistency between neighboring warped images with a self-supervised loss, reducing the need for strong supervision.

We first review related work in weak temporal and photometric supervision Section 4.1. We then describe our proposed model and method for leveraging photometric consistency

over time in Section 4.1.2. Finally, we present empirical evidence to show the strength of our proposed baseline and temporal supervision approach in Section 4.3.

4.1 Related work

In RGB videos, motion cues provide useful information that can be used for self-supervision. Several methods explore this idea in the context of human body pose estimation. None of these methods focuses on hands, and more particularly on complex hand-object interactions.

4.1.1 Learning with temporal constraints

Pfister et al. (2015) leverage optical flow for 2D human pose estimation. Slim DensePose Neverova et al. (2019) uses an off-the-shelf optical flow method Ilg et al. (2017) to establish dense correspondence Guler et al. (2018) between adjacent frames in a video. These correspondences are used to propagate manual annotations between frames and to enforce spatio-temporal equivariance constraints. Very recently, PoseWarper Bertasius et al. (2019) leverages image features to learn the pose warping between a labeled frame and an unlabeled one, thus propagating annotations in sparsely labeled videos.

Regressing 3D poses is more difficult: the problem is fundamentally ambiguous in monocular scenarios. Furthermore, collecting 3D annotations is not as easy as in 2D. VideoPose3D Pavllo et al. (2019) regresses 3D skeleton joint positions, by back-projecting them on the image space and using CNN-estimated 2D keypoints as supervision. Tung et al. (2017) regress the SMPL body model parameters Loper et al. (2015) and use optical flow and reprojected masks to provide weak supervision. Differently from us, they rely on an off-the-shelf optical flow method, making the pose accuracy dependent on the flow quality. Recently, Arnab et al. (2019) refine noisy per-frame pose predictions Kanazawa et al. (2018a) using bundle adjustment over the SMPL parameters in a video clip.

4.1.2 Learning with photometric consistency

Our method enforces photometric consistency between pose estimates from adjacent frames. Similar ideas have been successfully applied to self-supervised learning of ego-motion, depth and scene flow for self-driving cars Brickwedde et al. (2019); Godard et al. (2017); Zhou et al. (2017). Unlike these methods, which estimate pixel-wise probability depth distributions for mostly rigid scenes, we focus on estimating the articulated pose of hands manipulating objects. Starting from multi-view setups at training time, Rhodin et al. (2018a,b) propose weak supervision strategies for monocular human pose estimation. We consider

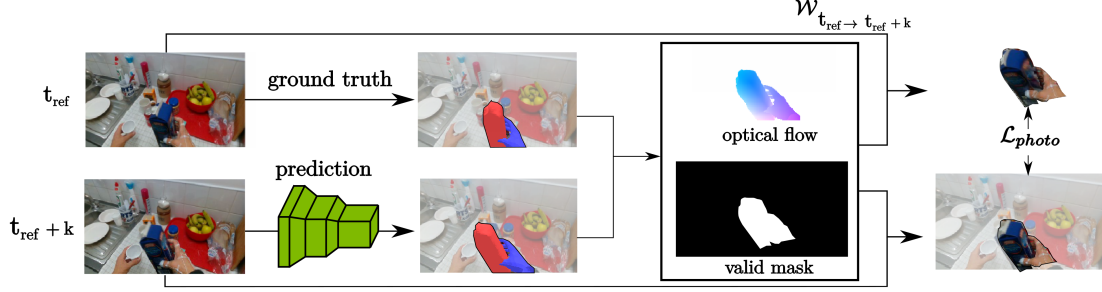


Figure 4-1: Photometric consistency loss. Given an annotated frame, t_{ref} , and an unannotated one, t_{ref+k} , we reconstruct hand and object 3D pose at t_{ref+k} leveraging a self-supervised loss. We differentiably render the optical flow between ground-truth hand-object vertices at t_{ref} and estimated ones. Then, we use this flow to warp frame t_{ref+k} into t_{ref} , and enforce consistency in pixel space between warped and real image.

monocular setups where the camera might move. Similarly to us, TexturePose Pavlakos et al. (2019b) enforces photometric consistency between pairs of frames to refine body pose estimates. They define the consistency loss in UV space: this assumes a UV parameterization is always provided. Instead, we define our loss in image space. Notably, these methods consider scenarios without severe occlusions (only one instance, one body, is in the scene).

4.2 Learning to grasp known objects with sparse temporal supervision

We propose a CNN-based model for 3D hand-object reconstruction that can be efficiently trained from a set of *sparse annotated* video frames. Differently from Chapter 3, we follow previous work Kehl et al. (2017); Tekin et al. (2019, 2018) in assuming that a 3D mesh model of the object is provided. The key idea of our approach is to use a photometric consistency loss, that we leverage as self-supervision on the unannotated intermediate frames in order to improve hand-object reconstructions. We introduce this loss in Section 4.2.1. We then describe our learning framework in detail in Section 4.2.2.

4.2.1 Temporal supervision from sparse grasp supervision

In the following, we describe our photometric consistency loss which leverages temporal consistency across neighboring frames to provide weak supervision during training as described in Section 4.2.2.

3D-aware temporal consistency. As mentioned above, our method takes as input a sequence of RGB frames and outputs hand and object mesh vertex locations for each frame. The same type of output is generated in 3, where each RGB frame is processed separately. We observe that the temporal continuity in videos imposes temporal constraints between neighboring frames. We assume that 3D annotations are provided only for a sparse subset of frames; this is a scenario that occurs in practice when data collection is performed on sequential images, but only a subset of them is manually annotated. We then define a self-supervised loss to propagate this information to unlabeled frames. Our self-supervised loss exploits photometric consistency between frames, and is defined in image space. Figure 4-1 illustrates the process.

Temporal consistency intuition. Consider an annotated frame $I_{t_{ref}}$ at time t_{ref} , for which we have ground-truth hand and object vertices $V_{t_{ref}}$ (to simplify the notation, we do not distinguish here between hand and object vertices). Given an unlabeled frame $I_{t_{ref}+k}$, our goal is to accurately regress hand and object vertex locations $V_{t_{ref}+k}$. Our main insight is that, given estimated per-frame 3D meshes and known camera intrinsic parameters, we can back-project our meshes on image space and leverage pixel-level information to provide additional cross-frame supervision. Given $I_{t_{ref}+k}$, we first regress hand and object vertices $V_{t_{ref}+k}$ in a single feed-forward network pass (see Sec. 4.2.2). Imagine now to back-project these vertices on $I_{t_{ref}+k}$ and assign to each vertex the color of the pixel they are projected onto. The object meshes at t_{ref} and $t_{ref}+k$ share the same topology; and so do the hand meshes. So, if we back-project the ground-truth meshes at t_{ref} on $I_{t_{ref}}$, corresponding vertices from $V_{t_{ref}}$ and $V_{t_{ref}+k}$ should be assigned the same color, up to changes due to lighting and occlusions.

Photometric consistency loss. We translate this idea into our photometric consistency loss. We compute the 3D displacement (“flow”) between corresponding vertices from $V_{t_{ref}}$ and $V_{t_{ref}+k}$. These values are then projected on the image plane, and interpolated on the visible mesh triangles. To this end, we differentiably render the estimated flow from $V_{t_{ref}}$ to $V_{t_{ref}+k}$ using the Neural Renderer Kato et al. (2018). This allows us to define a warping flow $\mathcal{W}_{t_{ref}+k \rightarrow t_{ref}}$ between the pair of images as a function of $V_{t_{ref}+k}$ and $V_{t_{ref}}$.

We exploit the computed flow to warp $I_{t_{ref}+k}$ into the warped image $\mathcal{W}_{t_{ref}+k \rightarrow t_{ref}}(I_{t_{ref}+k}, V_{t_{ref}}, V_{t_{ref}+k})$, by differentiably sampling values from $I_{t_{ref}+k}$ according to the predicted optical flow displacements, which is computed using $V_{t_{ref}}$ and $V_{t_{ref}+k}$. Our loss enforces consistency between the warped image and the reference one. Note that the error is minimized with respect to the estimated hand and object vertices $V_{t_{ref}+k}$.

$$\mathcal{L}_{photo}(V_{t_{ref}+k}) = ||M \cdot (\mathcal{W}(I_{t_{ref}+k}, V_{t_{ref}}, V_{t_{ref}+k}) - I_{t_{ref}})||_1, \quad (4.1)$$

where M is a binary mask denoting surface point visibility. In order to compute the visibility mask, we ensure that the supervised pixels belong to the silhouette of the reprojected mesh in the target frame $I_{t_{ref}+k}$. We additionally verify that the supervision is not applied to pixels which are occluded in the reference frame by performing a cyclic consistency check.

Cycle consistent visibility check. Our consistency check is similar to Hur and Roth (2017); Neverova et al. (2019). Let us denote the flow warping the estimated frame $I_{t_{ref}+k}$ into the reference one $I_{t_{ref}}$ by $\mathcal{W}_{t_{ref}+k \rightarrow t_{ref}}$. Similarly, we compute a warping flow in the opposite direction, from the reference frame to the estimated one: $\mathcal{W}_{t_{ref} \rightarrow t_{ref}+k}$. Given the mask $M_{t_{ref}}$ obtained by projecting $V_{t_{ref}}$ on image space, we consider each pixel $p \in M_{t_{ref}+k}$. We warp p into the reference frame, and then back into the estimated one: $\tilde{p} = \mathcal{W}_{t_{ref}+k \rightarrow t_{ref}}(\mathcal{W}_{t_{ref} \rightarrow t_{ref}+k}(p))$. If the distance between p and \tilde{p} is greater than 2 pixels, we do not apply our loss at this location. On FPHAB, when using 1% of the data as reference frames, this check discards 3.3% of $M_{t_{ref}+k}$ pixels.

We successively warp a grid of pixel locations using the optical flow t_{ref} to $t_{ref} + k$ and from $t_{ref} + k$ to t_{ref} and include only pixel locations which remain stable, a constraint which does not hold for mesh surface points which are occluded in one of the frames.

Differences to previous work. The consistency supervision \mathcal{L}_{photo} can be applied directly on pixels, similarly to self-supervised ego-motion and depth learning scenarios Godard et al. (2017); Zhou et al. (2017). The main difference with these approaches is that they estimate per-pixel depth values while we attempt to leverage the photometric consistency loss in order to refine rigid and articulated motions. Our approach is similar in spirit to that of Pavlakos et al. (2019b). With respect to them, we consider a more challenging scenario (multiple 3D instances and large occlusions). Furthermore, we define our loss in image space, instead of UV space, and thus we do not assume that a UV parametrization is available.

As each operation is differentiable, we can combine this loss and use it as supervision either in isolation or in addition to other reconstruction losses (Sec. 4.2.2).

4.2.2 Learning to grasp known objects

We apply the loss introduced in Section 4.2.1 to 3D hand-object reconstructions obtained independently for each frame. These per-frame estimates are obtained with a single for-

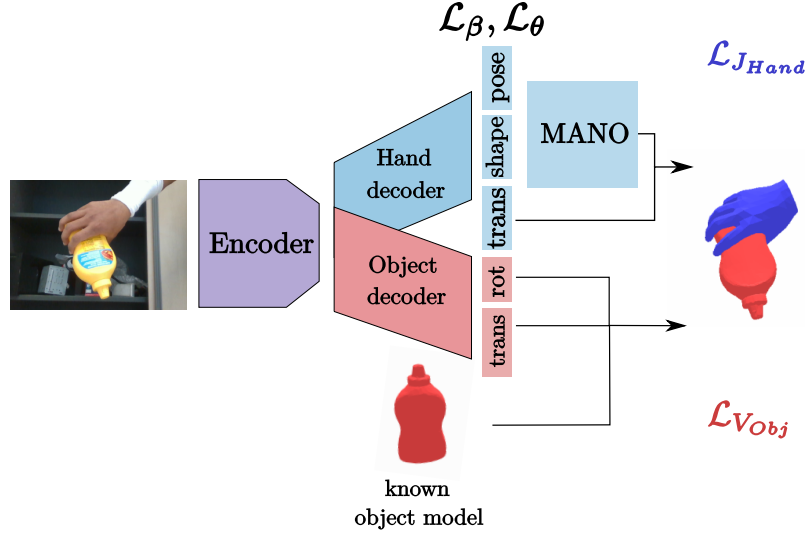


Figure 4-2: Architecture of the single-frame hand-object reconstruction network. Our network assumes that the object CAD model is available and regresses pose parameters for the hand and object as well as MANO hand shape parameters.

ward pass through a deep neural network, whose architecture is shown in Fig. 4-2. In the following, we detail our proposed method to regress the hand-object configuration from RGB inputs 4.2.2 and the training details 4.2.2.

Hand-object reconstruction with known object model.

Network architecture. In the spirit of Chapter 3 and Boukhayma et al. (2019), our network takes as input a single RGB image and outputs MANO Romero et al. (2017) pose and shape parameters. However, differently from Chapter 3, we assume that a 3D model of the object is given, and we regress its 6D pose by adding a second head to our network (see again Fig. 4-2). We employ as backbone a simple ResNet-18 He et al. (2015), which is

Branch	Input shape	Output shape	ReLU
Object pose regressor	512 256	256 6	✓
Hand translation regressor	512 256	256 3	✓
Hand pose and shape regressor	512 512 512	512 512 28	✓ ✓

Table 4.1: **Architecture of the Hand and Object parameter regression branches.** We use fully connected linear layers to regress pose and shape parameters from the 512-dimensional features.

computationally efficient. We use the base network model as the image encoder and select the last layer before the classifier to produce our image features. We then regress hand and object parameters from these features through 2 dense layers with ReLU non-linearities. We extract image features from the last layer of ResNet18 before softmax. We regress in separate branches 6 parameters for the global object translation and rotation, 3 parameters for the global hand translation, and 28 MANO parameters which account for global hand rotation, articulated pose and shape deformation. The details of each branch are presented in Table 4.1.

Hand-object *global* pose estimation. We formulate the hand-object global pose estimation problem in the camera coordinate system and aim to find precise absolute 3D positions of hands and objects. Instead of a weak perspective camera model, commonly used in the body pose estimation literature, we choose here to use a more realistic projective model. In our images, hand-object interactions are usually captured at a short distance from the camera. So the assumptions underlying weak perspective models do not hold. Instead, we follow best practices from object pose estimation. As in Li et al. (2018); Xiang et al. (2018), we predict values that can be easily estimated from image evidence. Namely, in order to estimate hand and object translation, we regress a focal-normalized depth offset d_f and a 2D translation vector (t_u, t_v) , defined in pixel space. We compute d_f as

$$d_f = \frac{V_z - z_{off}}{f}, \quad (4.2)$$

where V_z is the distance between mesh vertex and camera center along the z-axis, f is the camera focal length, and z_{off} is empirically set to $40cm$. t_u and t_v represent the translation, in pixels, of the object (or hand) origin, projected on the image space, with respect to the image center. Note that we regress d_f and (t_u, t_v) for both the hand and the object, separately.

Given the estimated d_f and (t_u, t_v) , and the camera intrinsic parameters, we can easily derive the object (hand) global translation in 3D. For the global rotation, we adopt the axis-angle representation. Following Kanazawa et al. (2018a); Li et al. (2018); Pavlakos et al. (2018), the rotation for object and hand is predicted in the object-centered coordinate system.

Articulated hand pose and shape estimation. We obtain hand 3D reconstructions by predicting MANO pose and shape parameters. For the pose, similarly to Chapter 3 and Boukhayma et al. (2019), we predict the principal component analysis (PCA) coefficients of the low-dimensional hand pose space provided in Romero et al. (2017). For the shape, we predict the MANO shape parameters, which control identity-specific characteristics such as skeleton bone length. Overall, we predict 15 pose coefficients and 10 shape parameters.

Reconstruction losses. In total, we predict 6 parameters for hand-object rotation and translation and 25 MANO parameters, which result in a total of 37 regressed parameters. We then apply the predicted transformations to the reference hand and object models and further produce the 3D joint locations of the MANO hand model, which are output by MANO in addition to the hand vertex locations. We define our supervision on hand joint positions, $\mathcal{L}_{J_{Hand}}$, as well as on 3D object vertices, $\mathcal{L}_{V_{Obj}}$. Both losses are defined as ℓ_2 errors.

Regularization losses. We find it effective to regularize both hand pose and shape by applying ℓ_2 penalization as in Boukhayma et al. (2019). $\mathcal{L}_{\theta_{Hand}}$ prevents unnatural joint rotations, while $\mathcal{L}_{\beta_{Hand}}$ prevents extreme shape deformations, which can result in irregular and unrealistic hand meshes.

Our final loss \mathcal{L}_{HO} is a weighted sum of the reconstruction and regularization terms:

$$\mathcal{L}_{HO} = \mathcal{L}_{V_{Obj}} + \lambda_J \mathcal{L}_{J_{Hand}} + \lambda_\beta \mathcal{L}_{\beta_{Hand}} + \lambda_\theta \mathcal{L}_{\theta_{Hand}}. \quad (4.3)$$

Skeleton adaptation. Hand skeleton models can vary substantially between datasets, resulting in inconsistencies in the definition of joint locations. Skeleton mismatches may force unnatural deformations of the hand model. To account for these differences, we replace the fixed MANO joint regressor with a skeleton adaptation layer which regresses joint locations from vertex positions. We initialize this linear regressor using the values from the MANO joint regressor and optimize it jointly with the network weights. We keep the tips of the fingers and the wrist joint fixed to the original locations, and learn a dataset-specific mapping for the other joints at training time. The positive effect of skeleton adaptation is presented in Subsection 4.3.4.

Training. All models are trained using the PyTorch Paszke et al. (2019) framework. We use the Adam Kingma and Ba (2014) optimizer with a learning rate of $5 \cdot 10^{-5}$. We initialize the weights of our network using the weights of a ResNet He et al. (2015) trained on ImageNet Deng et al. (2009). We empirically observed improved stability during training when freezing the weights of the batch normalization Ioffe and Szegedy (2015) layer to the weights initialized on ImageNet.

We pretrain the models on fractions of the data without the consistency loss. As an epoch contains fewer iterations when using a subset of the dataset, we observe that a larger number of epochs is needed to reach convergence for smaller fractions of training data. We later fine-tune our network with the consistency loss using a fixed number of 200 epochs.

4.2.3 Dense 3D Hand-Object Reconstruction

4.3 Experiments

In this section, we first describe the datasets in Section 4.3.1 and corresponding evaluation protocols. We then recall the used Section 4.3.2 and compare our method to the state of the art and provide a detailed analysis of our framework in Section 4.3.3. Finally, we validate numerically and qualitatively the use of our learnt skeleton adaptation layer in Section 4.3.4 and runtime in Section 4.3.5.

4.3.1 Datasets

We evaluate our framework for joint 3D hand-object reconstruction and pose estimation on two recently released datasets: First Person Hand Action Benchmark Garcia-Hernando et al. (2018) and HO-3D Hampali et al. (2019) which provide pose annotations for all hand keypoints as well as the manipulated rigid object.

First-person hand action benchmark (FPHAB): The FPHAB dataset Garcia-Hernando et al. (2018) collects egocentric RGB-D videos capturing a wide range of hand-object interactions, with ground-truth annotations for 3D hand pose, 6D object pose, and hand joint locations. The annotations are obtained in an automated way, using mocap magnetic sensors strapped on hands. Object pose annotations are available for 4 objects, for a subset of the videos. Similarly to hand annotations, they are obtained via magnetic sensors. In our evaluation, we use the same *action split* as in Tekin et al. (2019): each object is present in both the training and test splits, thus allowing the model to learn instance-specific 6 degrees of freedom (DoF) transformations. To further compare our results to those of Chapter 3, we also use the *subject split* of FPHAB where the training and test splits feature different subjects.

HO-3D: The recent HO-3D dataset Hampali et al. (2020) is the result of an effort to collect 3D pose annotations for both hands and manipulated objects in a markerless setting. In this work, we report results on the subset of the dataset which was released as the first version Hampali et al. (2019). The subset of HO-3D we focus on contains 14 sequences, out of which 2 are available for evaluation. The authors augment the real training sequences with additional synthetic data. In order to compare our method against the baselines introduced in Hampali et al. (2019), we train jointly on their real and synthetic training sets.

In this Chapter, we work with the subset of the dataset which was first released. Out of the 68 sequences which have been released as the final version of the dataset, 15 have been

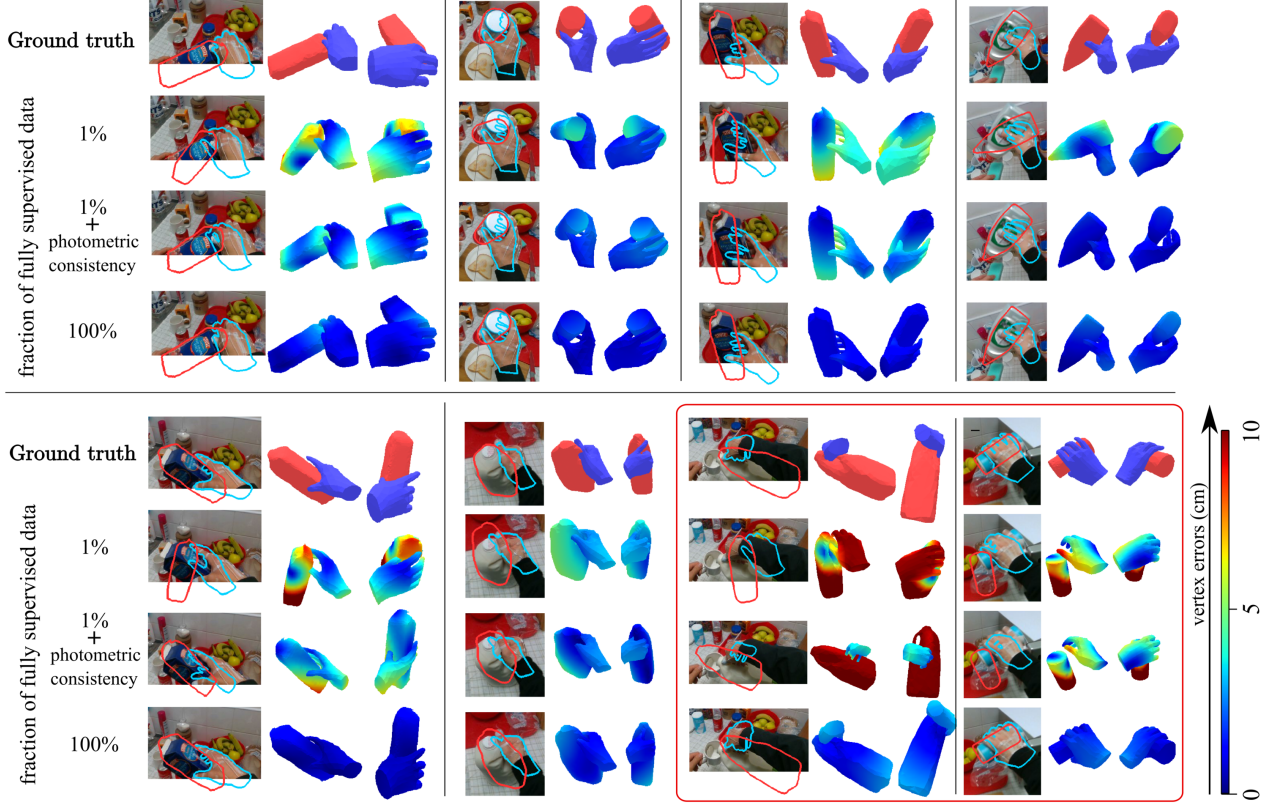


Figure 4-3: Qualitative results on the FPHAB dataset. We visualize the reconstructed meshes re-projected on the image as well as a rotated view. When training on the full dataset, we obtain reconstructions which accurately capture the hand-object interaction. In the sparsely supervised setting, we qualitatively observe that photometric consistency allows to recover more accurate hand and object poses. Failure cases occur in the presence of important motion blur and large occlusions of the hand or the object by the subject’s arm.

made available as part of an earlier release. Out of these, we select the 14 sequences that depict manipulation of two following objects: the mustard bottle and the cracker box. The train sequences in this subset are the ones named SM2, SM3, SM4, SM5, MC4, MC6, SS1, SS2, SS3, SM2, MC1, MC5. When experimenting with the photometric consistency, we use SM1 and MC2 as the two test sequences. When comparing to the baseline of Hampali et al. (2019), we use MC2 as the unique test sequence.

4.3.2 Evaluation Metrics

We evaluate our approach on 3D hand pose estimation and 6D object pose estimation and use official train/test splits to evaluate our performance in comparison to the state of the art. We report accuracy using the following metrics.

Method	Hand error	Object error
Tekin et al. (2019)	15.8	24.9
Ours	18.0	22.3

Table 4.2: Comparison to state-of-the-art method of Tekin et al. (2019) on FPHAB Garcia-Hernando et al. (2018), errors are reported in mm.

Mean 3D errors. To assess the quality of our 3D hand reconstructions, we compute the mean end-point error (in mm) over 21 joints following Zimmermann and Brox (2017). For objects, on FPHAB we compute the average vertex distance (in mm) in camera coordinates to compare against Tekin et al. (2019), on HO-3D, we look at average bounding box corner distances.

Mean 2D errors. We report the mean errors between reprojected keypoints and 2D ground-truth locations for hands and objects. To evaluate hand pose estimation accuracy, we measure the average joint distance. For object pose estimation, following the protocol for 3D error metrics, we report average 2D vertex distance on FPHAB, and average 2D corner distance on HO-3D. To further compare our results against Hampali et al. (2019), we also report the percentage of correct keypoints (PCK). To do so, for different pixel distances, we compute the percentage of frames for which the average error is lower than the given threshold.

4.3.3 Experimental Results

We first report the pose estimation accuracy of our single-frame hand-object reconstruction model and compare it against the state of the art Hampali et al. (2019); Tekin et al. (2019). We then present the results of our motion-based self-supervised learning approach and demonstrate its efficiency in case of scarcity of ground-truth annotations.

Single-frame hand-object reconstruction. Taking color images as input, our model reconstructs dense meshes to leverage pixel-level consistency, and infers hand and object poses.

SOTA comparison on FPHAB. To compare our results to the state of the art Hampali et al. (2019); Tekin et al. (2019) and Chapter 3, we evaluate our pose estimation accuracy on the FPHAB Garcia-Hernando et al. (2018) and HO-3D Hampali et al. (2019) datasets. Table 4.2 demonstrates that our model achieves better accuracy than Tekin et al. (2019) on object pose estimation. We attribute this to the fact that Tekin et al. (2019) regresses keypoint positions, and recovers the object pose as a non-differentiable post-processing

	Hand error (mm)	Object error (mm)
Hand only	15.7	-
Object only	-	21.8
Hand + Object	18.0	22.3

Table 4.3: We compare training for hand and object pose estimation jointly and separately on FPHAB Garcia-Hernando et al. (2018) and find that the encoder can be shared at a minor performance cost in hand and object pose accuracy.

step, while we directly optimize for the 6D pose. Our method achieves on average a hand pose estimation error of 18 mm on FPHAB which is outperformed by Tekin et al. (2019) by a margin of 2.6 mm. This experiment is in line with earlier reported results, where the estimation of individual keypoint locations outperformed regression of model parameters Kanazawa et al. (2018a); Pavlakos et al. (2019b, 2018). While providing competitive pose estimation accuracy to the state of the art, our approach has the advantage of predicting a detailed hand shape, which is crucial for fine-grained understanding of hand-object interactions and contact points. We further compare our results to those of Chapter 3 that reports results on FPHAB using the *subject split* and demonstrate that our model provides improved hand pose estimation accuracy, while additionally estimating the global position of the hand in the camera space.

Quantitative comparison on HO-3D.

We further evaluate the hand-object pose estimation accuracy of our single-image model on the recently introduced HO-3D dataset. We show in Fig. 4-4 that we outperform Hampali et al. (2019) on both hand and object pose estimation. In Table 4.3, we analyze the effect of simultaneously training for hand and object pose estimation within a unified framework. We compare the results of our unified model to those of the models trained individually for hand pose estimation and object pose estimation. We observe that the unified co-training slightly degrades hand pose accuracy. This phenomenon is also observed by Tekin et al. (2019), and might be due to the fact that while the hand pose highly constrains the object pose, simultaneous estimation of the object pose does not result in increased hand pose estimation accuracy, due to higher degrees of freedom inherent to the articulated pose estimation problem.

Photometric supervision on video. We now validate the efficiency of our self-supervised dense hand-object reconstruction approach when ground-truth data availability is limited. We pretrain several models on a fraction of the data by sampling frames uniformly in each sequence. We sample a number of frames to reach the desired ratio of annotated frames in each training video sequence, starting from the first frame. We then continue training with

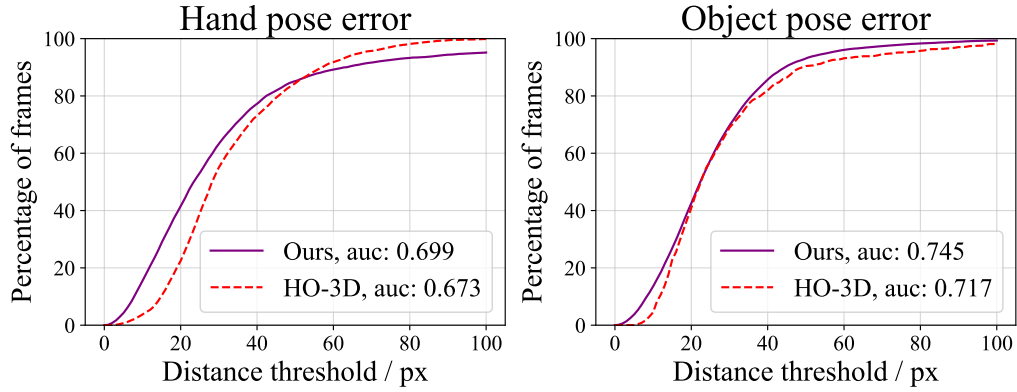


Figure 4-4: Evaluation of our baseline for hand-object pose estimation on the early release of the HO-3D Hampali et al. (2019) dataset. We report the PCK for 2D joint mean-end-point error for hands, and the mean 2D reprojection error for objects.

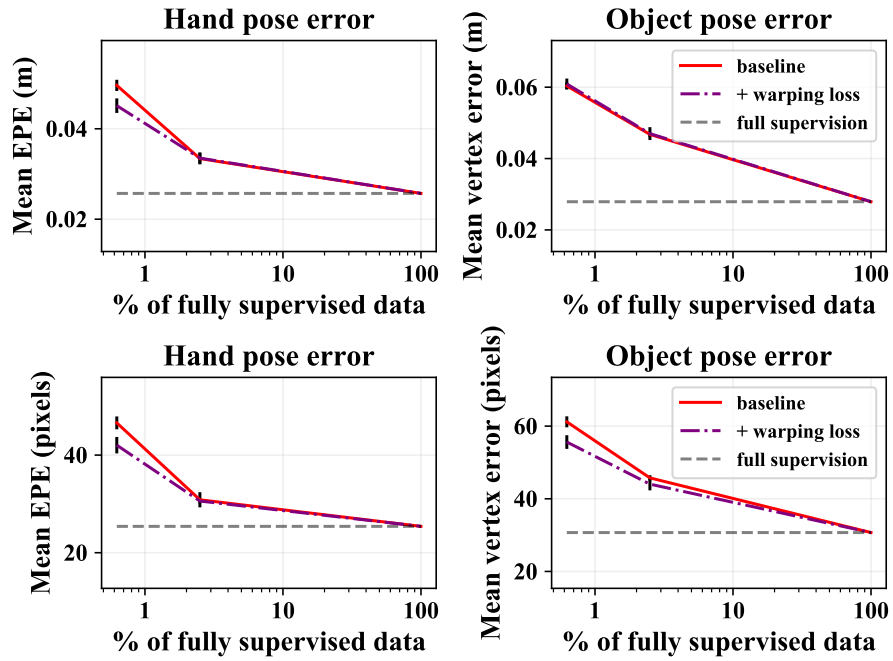


Figure 4-5: Effect of using photometric-consistency self-supervision when only a fraction of frames are fully annotated on HO-3D. We report average values and standard deviations over 5 different runs.

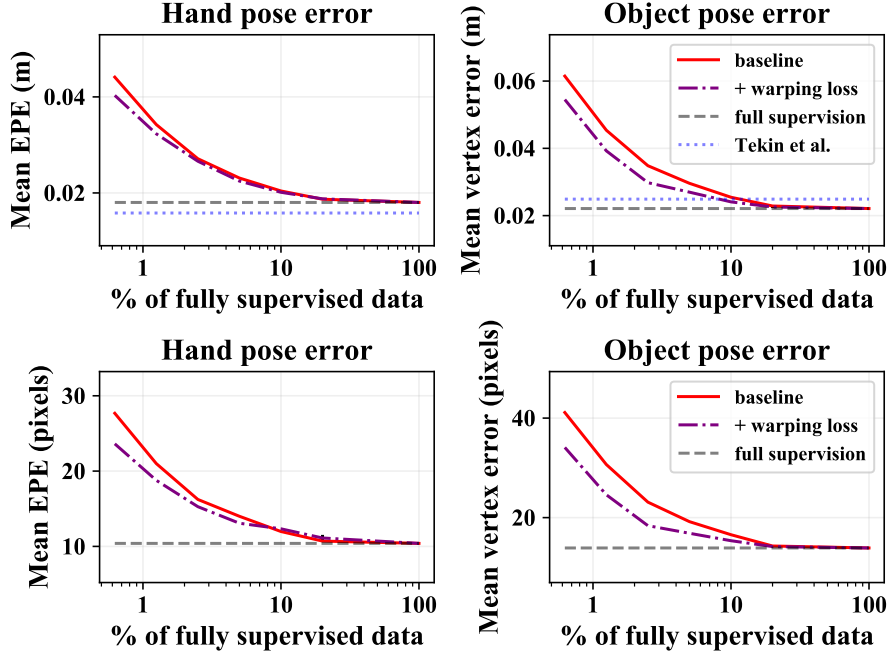


Figure 4-6: We observe consistent quantitative improvements from the photometric consistency loss as the percentage of fully supervised frames decreases below 10% for both hands and objects.

photometric consistency as an additional loss, while maintaining the full supervision on the sparsely annotated frames. Additional implementation and training details are discussed in Section 4.2.2. In order to single out the effect of the additional consistency term and factor out potential benefits from a longer training time, we continue training a reference model with the full supervision on the sparse keyframes for comparison. We experiment with various regimes of data scarcity, progressively decreasing the percentage of annotated keyframes from 50 to less than 1%.

We report our results in Fig. 4-6 for FPHAB and in Fig. 4-5 for HO-3D. We observe that only 20% of the frames are necessary to reach the densely supervised performance on the FPHAB dataset, which can be explained by the correlated nature between neighboring frames. However, as we further decrease the fraction of annotated data, the generalization error significantly decreases. We demonstrate that our self-supervised learning strategy significantly improves the pose estimation accuracy in the low data regime when only a few percent of the actual dataset size are annotated and reduces the rigid reliance on large labeled datasets for hand-object reconstruction. Although the similarity between the reference and consistency-supervised frames decreases as the supervision across video becomes more sparse and the average distance to the reference frame increases, resulting in larger appearance changes, we observe that the benefits from our additional photometric con-

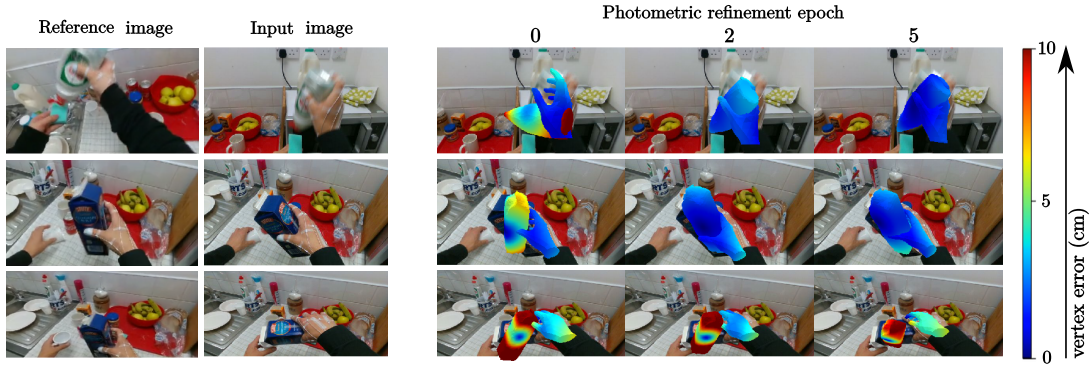


Figure 4-7: Progressive pose refinement over training samples, even in the presence of large motion and inaccurate initialization. In extreme cases (last row), the model cannot recover.

sistency is most noticeable for both hands and objects as scarcity of fully annotated data increases. When using less than one percent of the training data with full supervision, we observe an absolute average improvement of 7 pixels for objects and 4 pixels for hands, reducing the gap between the sparsely and fully supervised setting by respectively 25 and 23% (see Fig. 4-6). While on HO-3D the pixel-level improvements on objects do not translate to better 3D reconstruction scores for the object (see Fig. 4-5), on FPHAB, the highest relative improvement is observed for object poses when fully supervising 2.5% of the data. In this setup, the 4.7 reduction in the average pixel error corresponds to a reduction of the error by 51% and results in a reduction by 40% in the 3D *mm* error. We qualitatively investigate the modes of improvement and failure from introducing the additional photometric consistency loss in Figure 4-3 and Figure 4-8.

As our method relies on photometric consistency for supervision, it is susceptible to fail when the photometric consistency assumption is infringed, which can occur for instance in cases of fast motions or illumination changes. However, our method has the potential to provide meaningful supervision in cases where large motions occur between the reference and target frames, as long as the photometric consistency hypothesis holds. We observe that in most cases, our baseline provides reasonable initial pose estimates on unannotated frames, which allows the photometric loss to provide informative gradients. In Figure 4-7, we show examples of successful and failed pose refinements on training samples from the FPHAB dataset supervised by our loss. Our model is able to improve pose estimations in challenging cases, where the initial prediction is inaccurate and there are large motions with respect to the reference frame.

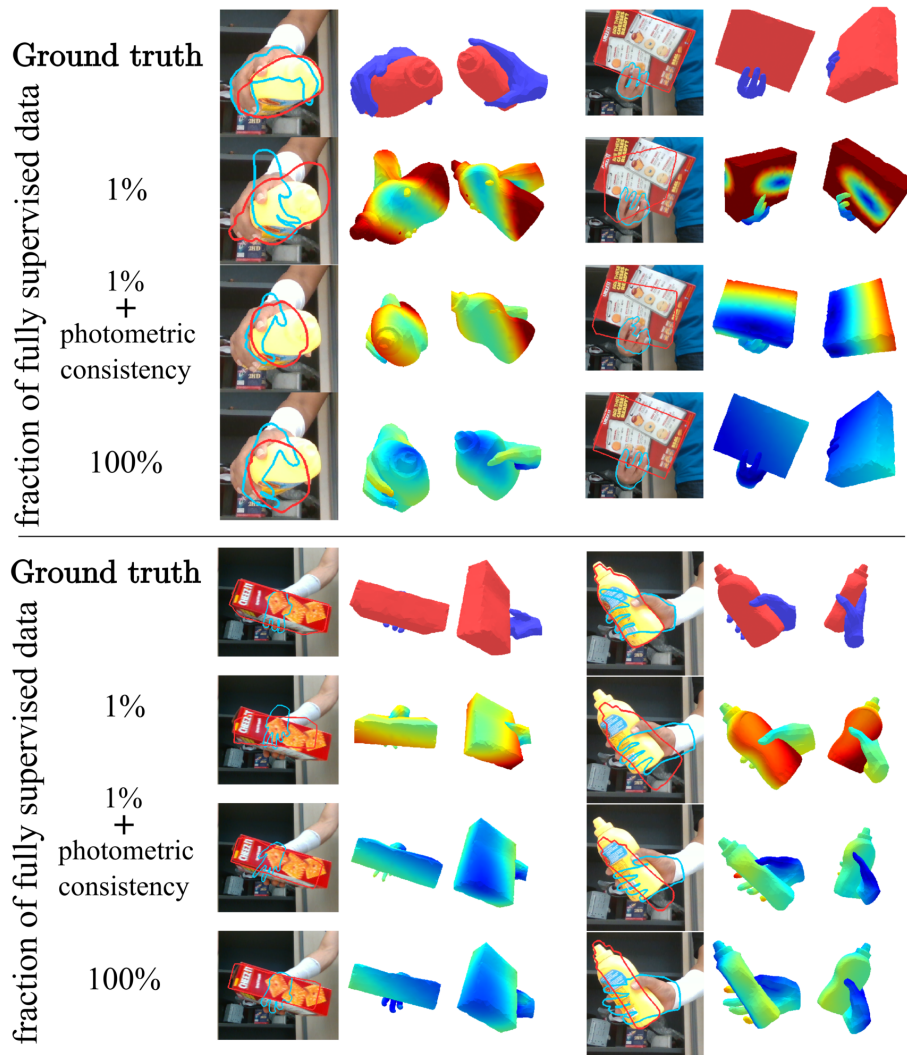


Figure 4-8: Predicted reconstructions for images from HO-3D. While rotation errors around axis parallel to the camera plane are not corrected and are sometimes even introduced by the photometric consistency loss, we observe qualitative improvement in the 2D reprojection of the predicted meshes on the image plane.

Method	Hand error
Ours - no skeleton adaptation	28.1
Ours	27.4
Chapter 3	28.0

Table 4.4: On the FHPAB dataset, for which the skeleton is substantially different from the MANO one, we show that adding a skeleton adaptation layer allows us to outperform our results from Chapter 3, while additionally predicting the global translation of the hand.

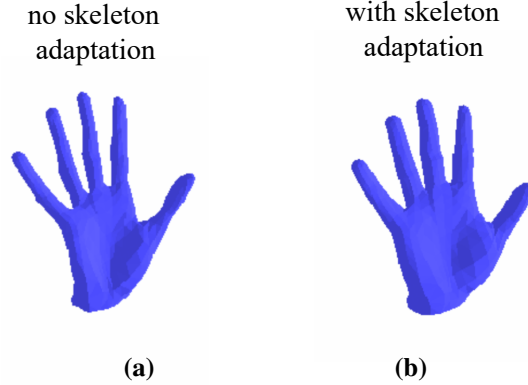


Figure 4-9: Predicted shape deformations in the (a) absence and (b) presence of the skeleton adaptation layer on the FPHAB dataset.

4.3.4 Skeleton adaptation

The defined locations for the joints do not exactly match each other for the FPHAB Garcia-Hernando et al. (2018) dataset and the MANO Romero et al. (2017) hand model.

Quantitative evaluation. As shown in Table 4.4, we observe marginal improvements in the average joint predictions using our skeleton adaptation layer. This demonstrates that MANO has already the ability to deform sufficiently to account for various skeleton conventions. However, these deformations come at the expense of the realism of the reconstructed meshes, which undergo unnatural deformations in order to account for the displacements of the joints. To demonstrate this effect, we train a model on the FPHAB Garcia-Hernando et al. (2018) dataset, without the linear skeleton adaptation layer, and qualitatively compare the predicted hand meshes with and without skeleton adaptation.

Qualitative analysis. We observe in Fig. 4-9(a) that, without skeleton adaptation, the fingers get unnaturally elongated to account for different definitions of the joint locations in FPHAB and MANO. As shown in Fig. 4-9(b), we are able to achieve higher realism for the reconstructed meshes using our skeleton adaptation layer.

4.3.5 Runtime.

The forward pass runs in real time, at 34 frames per second on a Titan X GPU.

4.4 Conclusions

In this chapter, we proposed a new method for dense 3D reconstruction of hands and objects from monocular color images. We presented a sparsely supervised learning approach

leveraging photo-consistency between sparsely supervised frames. We demonstrated that our approach achieves high accuracy for hand and object pose estimation and successfully leverages similarities between sparsely annotated and unannotated neighboring frames to provide additional supervision.

Our work relies on a simple pixel-level loss for weak supervision, which makes it easy to implement but has limited robustness to photometric variations which can come from changing lighting between two frames, or occlusions not modeled by our reconstruction, such as the ones which are due to the passive environment. Several improvements could contribute to improving the proposed method. In particular, a more robust loss would benefit the system and likely result in a stronger signal from the annotated frame’s temporal neighborhood. Our framework is general and can be extended to incorporate the full 3D human body along with the environment surfaces, which would allow to more exhaustively account for possible occlusions, and thus result in improved supervision. While our current model requires precise annotations for the reference frames, these improvements could allow for some noise, which would make it more widely and practically applicable.

Chapter 5

Joint hand-object fitting

In the previous chapters, we developed learning-based methods which assume that accurate 3D ground truth is available at least for a subset of the training frames. However, in practice, scalable methods to annotate hand-object interactions are not available, preventing learning-based approaches from generalizing beyond their specific training domains. Developing an RGB-only method to retrieve 3D hand-object configurations would enable scaling up the datasets, and help the field move towards in-the-wild scenes.

In this chapter, we argue for an optimization-based approach for its robustness across domains. Recent progress in 2D detection of objects and 3D pose estimation of isolated hands makes it possible to obtain a good initialization when fitting 3D hand-object poses to these estimates. Nevertheless, this is still very challenging due to depth ambiguities, occlusions, noisy 2D estimates and physically implausible configurations.

We explore fitting hand and object meshes during interactions, methods which have so far been applied for automatic or hybrid data annotation from RGB-D or multi-view inputs, to RGB-only videos. Given that we use noisy predictions to guide our joint optimization, we investigate the importance of the accuracy of the object and hand model and find that this is an important component for a satisfactory performance. Our contributions are the following: (i) We propose a fitting-based approach for hand-object reconstruction from a video clip; (ii) We present a detailed quantitative evaluation analyzing different components of our optimization method and compare to learning-based models on a standard benchmark Hampali et al. (2019); (iii) We demonstrate qualitatively the capabilities of our framework to generalize on unconstrained videos; (iv) Finally, we show the benefits of using our fits as automatic labels to perform test-time training, otherwise referred to as self-labeling.

5.1 Related work

In the following, we provide an overview of methods which are related to the task of recovering the hand and object pose from RGB frames through joint fitting. We first review methods which have been deployed to annotate arbitrary objects in diverse RGB images in Section 5.1.1. We then review work that recovers noisy 3D fits from RGB data in the context of human action in Section 5.1.2. Finally, we provide references for methods which use temporal constraints to regularize estimated poses in Section 5.1.3.

5.1.1 Annotating 3D objects in real images.

Annotating 3D objects in diverse images is a challenging task. Several approaches were developed to annotate objects in diverse color images, resulting in datasets of increased sizes and diversity. All existing methods to align 3D models in single-view RGB images require manual work for each annotated image.

Satkin et al. (2018) turn to specialized software to build 3D scenes. They use Google SketchUp to align 3D models from the Google 3D Warehouse to furniture from images in the SUN database Xiao et al. (2010). Others develop dedicated software to facilitate the annotation task. Lim et al. (2013) annotate the IKEA dataset using a labeling tool developed for this purpose. A user localizes object keypoints for one of 225 known furniture model in 800 color images to estimate the pose of the object in the image. The interface renders the posed object to help the annotator validate or edit the selected correspondences. Xiang et al. (2014) create a manual annotation interface to annotate viewpoints for objects from 12 object categories and use it to annotate the PASCAL3D+ dataset. Their interface allows the user to first indicate the orientation of a target object model and subsequently match 2D locations for a set of 3D keypoints. Sun et al. (2018) extend the IKEA dataset from Lim et al. (2013): they use the same 3D models but curate annotations for 14600 images, a significantly larger number than in the original dataset. Similarly to Lim et al. (2013), they click keypoints in images and minimize the reprojection error to estimate the final object pose. While Lim et al. (2013) directly optimize for the final object pose using Levenberg-Marquardt Levenberg (1944); Marquardt (1963), Sun et al. (2018) first estimate the pose by searching through a discrete set of possible focal lengths, apply EPnP Lepetit et al. (2009) for each candidate and keep the solution with the lowest reprojection error. They subsequently refine the estimated pose with the Levenberg-Marquardt algorithm Levenberg (1944); Marquardt (1963) for 50 small random perturbations starting from the initial solution, keeping the solution with the minimal reprojection error as their final estimate. Xiang et al. (2016) annotate 90127 images from ImageNet Deng et al. (2009) with 44147 3D models from 100 categories from the ShapeNet Chang et al. (2015) dataset. They use a

common learnt embedding Song et al. (2016) for synthetic and real images to automatically retrieve 3D object model candidates for a color image. An annotator manually chooses the best candidate and uses a custom interface to align the model to the visual evidence by adjusting the 3D orientation as well as the in-plane rotation and zoom for the model overlayed on the target image. Dai et al. (2017) scan rooms using RGB-D sensors and asked workers to annotate the scene with posed CAD models from ShapeNet (Chang et al. (2015)) by allowing them to resize, translate and orient them to match the reconstructed 3D scene. All the above methods are typically time-consuming even when optimized interfaces are developed to accelerate the annotation process.

5.1.2 Joint fitting to RGB frames.

Our choice to model hand-object interactions by optimizing 3D models to evidence collected from RGB frames is strongly influenced by the work from Zhang et al. (2020). Zhang et al. (2020) propose a method which models human-object interactions in RGB frames by fitting human and object models using noisy evidence from learnt components. In particular, they use object classes and segmentation masks predicted by Kirillov et al. (2019) to initialize the pose of candidate object meshes, and outputs from Joo et al. (2020) to initialize dense posed human meshes. They fit the object and human models jointly to recover a coherent scene, explicitly modeling interactions. In particular, they manually assign 3D object model parts to human parts with which they are likely to be in contact, and penalize distances between them during fitting. Their work recovers compelling though approximate reconstructions from single RGB frames.

Concurrent work of Cao et al. (2020b) extends the optimization-based body-object reconstruction method PHOSA Zhang et al. (2020) to perform hand-object fitting. While our method shares similar optimization components with Cao et al. (2020b), it differs by leveraging video data. We additionally incorporate our fits in an end-to-end framework as a self-labelling component and showcase two-hand object manipulations.

5.1.3 Temporal constraints for motion modeling.

In case of video inputs, temporal constraints have been used for body motion estimation in the context of neural networks Hossain and Little (2018); Kanazawa et al. (2019), or optimization Arnab et al. (2019); Peng et al. (2018). For hands, Cai et al. (2019) proposes a graph convolutional approach to learn temporal dependencies. Hampali et al. (2020) make use of a temporal consistency term when fitting hand-object configurations to RGB-D data. We explore a similar term to obtain temporally smooth fits to RGB data and initialize the optimization from the previous frame’s fit.

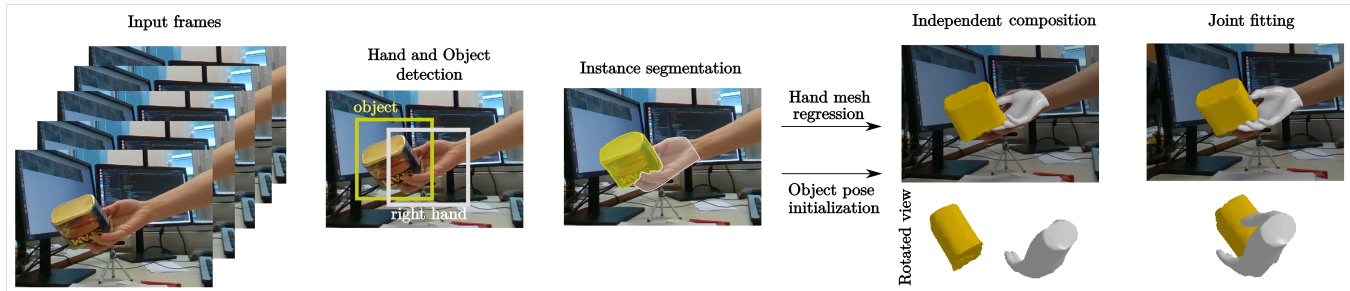


Figure 5-1: **Joint hand-object fitting:** We independently initialize the hand and object poses based on 2D detections and segmentations. We refine this configuration with interaction-based constraints to obtain our final joint fitting.

5.2 Fitting hand-object interactions in RGB images

We first describe the optimization-based fitting procedure, consisting of estimating 2D detections (Section 5.2.1), initializing 3D hand and object poses (Section 5.2.2), and joint fitting (Section 5.2.3). An overview can be seen in Figure 5-1.

Our method takes a video of hand-object manipulation as input. We assume that an exact or approximate object model representing the manipulated object is provided and use the ground-truth camera intrinsic parameters when available.

5.2.1 Obtaining 2D hand-object evidence

2D hand and object detection. For each video, the first step is to detect initial 2D bounding boxes. We use a recent hand and manipulated object detector Shan et al. (2020) to extract object and hand bounding boxes in each frame. The hands are predicted with left or right side labels. When the predicted boxes do not match the known properties of the dataset in terms of object presence, hand number or hand sides, we discard the detections for the given frame, and recover detections through tracking in a subsequent step.

Tracking. We apply an off-the-shelf 2D bounding box tracker Muron (2020) which relies on Kalman filtering Labbe (2014) to extract hand and object tracks from the noisy per-frame detections. This step allows to recover missed or discarded detections, and produces hand and object bounding box candidates for the full video clip.

For the Epic-Kitchens dataset, the real number of visible hands is unknown. We automatically select video clips for which at least one object and one hand track extend over more than 20 consecutive frames after tracking.

Segmentation. The key image evidence we rely on for fitting is 2D segmentation. We

extract instance masks $\hat{\mathcal{M}}_{obj}$ for each tracked object detection using the instance segmentation head of the PointRend Kirillov et al. (2019). Similar to PHOSA Zhang et al. (2020), we use a model pretrained on the COCO Lin et al. (2014) dataset. However, while Zhang et al. (2020) fits objects among the COCO categories, our target everyday objects are often not present among the COCO classes. For each object detection, we use the mask associated to the highest class activation of the PointRend instance classifier. We observe this class-agnostic approach to perform well in most cases. To account for hand occlusions, we extract the COCO masks associated to the *person* class $\hat{\mathcal{M}}_{hand}$ for the tracked hand boxes, see the supplemental material for additional details.

5.2.2 Independent pose initialization

Hand initialization. We employ the recent publicly available hand pose estimator FrankMocap Rong et al. (2020) to estimate the initial hand articulated poses, as well as the hand location and scale in pixel space. We recover an estimated depth using the world scale of the hand and the exact intrinsic camera parameters when available. When the exact camera intrinsic parameters are unknown, we approximate the focal length given the specifications of the camera, and assume the central point is at the center of the pixel image.

Object initialization. We use the 2D object segmentation to initialize the object pose for the 3D model associated to the target video clip. To obtain pose candidates for the first video frame, we sample random rotations uniformly in $\mathcal{SO}(3)$ and use the radius of the instance bounding box to estimate the 3D center of the provided mesh in the first frame. We optimize the object pose using differentiable rendering and a hand-occlusion aware silhouette error term following PHOSA Zhang et al. (2020). For each subsequent frame, we use the object pose from the previous frame as initialization. This process results in as many candidate motion initializations as there are candidate object poses. In practice, the number of candidate initializations is empirically set to 50.

We select the object motion candidate for which the average *IoU* score between the rendered mask and the target occlusion mask is highest.

5.2.3 Joint fitting

Independent hand-object fits are often inaccurate and do not take into account interaction-based constraints. We refine the initial hand-object poses leveraging both coarse and fine-grained manipulation priors.

Optimized parameters. The goal of our fitting is to find the optimal hand and object pose parameters for a sequence of T consecutive frames. For each frame, we optimize 3D

translations D_{hand} , D_{obj} , 3D global rotations R_{hand} , R_{obj} as well as θ hand pose parameters. Additionally, we optimize a shared hand scale s_{hand} .

We optimize the articulated MANO Romero et al. (2017) model in the latent pose space θ . Given the θ pose parameters, the MANO model differentiably outputs 3D hand vertex coordinates centered on the middle metacarpophalangeal joint $\mathcal{V}_{hand}^c = \text{MANO}(\theta)$. Following Taheri et al. (2020), we use a pose latent subspace of size 16. We optimize the hands and object rotation R_{hand} , R_{obj} using the 6D continuous rotation representation Zhou et al. (2019) and optimize the 3D translation D_{hand} , D_{obj} in metric camera space. When we use approximate object meshes, we additionally optimize a scalar scaling parameter s_{obj} which allows the object’s size to vary. We also allow hand vertices to scale by a factor s_{hand} which is shared across the T frames. The hand vertices in camera coordinates \mathcal{V}_{hand}^{3d} are estimated as following:

$$\mathcal{V}_{hand}^{3d} = s_{hand}(R_{hand}\mathcal{V}_{hand}^c) + D_{hand}. \quad (5.1)$$

The object vertices \mathcal{V}_{obj}^{3d} are estimated as a rigid transformation of canonically oriented model vertices \mathcal{V}_{obj}^c :

$$\mathcal{V}_{obj}^{3d} = s_{obj}(R_{obj}\mathcal{V}_{obj}^c) + D_{obj}. \quad (5.2)$$

Next, we describe the individual error terms that we minimize during fitting.

Object silhouette matching (\mathcal{L}_{obj}). We use a differentiable renderer Kato et al. (2018) to render the object mask \mathcal{M}_{obj} and compare it to the reference segmentation mask $\hat{\mathcal{M}}_{obj}$. This error term is occlusion-aware as in Zhang et al. (2020). No penalization occurs for the object silhouette being rendered in pixel regions $\hat{\mathcal{M}}_{hand}$ where hand occlusions occur. We write this error as:

$$\mathcal{L}_{obj} = ||(1 - \hat{\mathcal{M}}_{hand}) \circ (\mathcal{M}_{obj} - \hat{\mathcal{M}}_{obj})||_2^2 \quad (5.3)$$

Projected hand vertices (\mathcal{L}^{v2d}). We constrain the hand position by penalizing projected vertex offsets from the initial vertex pixel locations \mathcal{V}_{2d}^{hand} predicted by FrankMocap Rong et al. (2020). To compute the current 2D vertex locations, we project the MANO Romero et al. (2017) vertices \mathcal{V}_{hand}^{3D} to the pixel plane using the camera projection operation Π . This error is written as:

$$\mathcal{L}_{v2d} = ||\Pi(\mathcal{V}_{hand}^{3D}) - \hat{\mathcal{V}}_{hand}^{2d}||_2^2 \quad (5.4)$$

Hand regularization (\mathcal{L}_{pca}). Given that we optimize the *articulated* hand pose, we regularize the optimized hand pose. As in Boukhayma et al. (2019); Hasson et al. (2019); Rong et al. (2020), we apply a mean square error term to the PCA hand components $\mathcal{L}_{pca} = ||\theta||_2^2$ to bias the estimated hand poses towards statistically plausible configurations.

Scale (\mathcal{L}_{scale}). Similarly to PHOSAZhang et al. (2020), when we allow the elements in the scene to scale, we penalize deviations from category-level average dimensions.

Smoothness (\mathcal{L}_{smooth}). We further leverage a simple smoothness prior over the T sampled frames $\mathcal{L}_{smooth} = \sum_{t=1}^{T-1} \|\mathcal{V}_{t+1}^{3D} - \mathcal{V}_t^{3D}\|_2^2$ which encourages minimal 3D vertex variances across neighboring frames for both hands and objects as in Hampali et al. (2020).

Coarse interaction ($\mathcal{L}_{centroid}$). Following Zhang et al. (2020), we penalize the squared distance between hand and object centroids when the predicted hand and object boxes overlap to encode a coarse interaction prior. As we assume the object scale to be provided, this error only impacts the rigid hand pose, effectively attracting the hand towards the interacted object. In case of multiple hands, all overlapping hand-object pairs of meshes are considered.

Collision (\mathcal{L}_{col}). We rely on a recent collision penalization term introduced to enforce non-interpenetration constraints between multiple persons in the context of body mesh estimation Jiang et al. (2020). The collision error \mathcal{L}_{col} is computed for each pair k, l of estimated meshes. We compute $L_{col}^{k,l} = \sum_i \Phi_k(\mathcal{V}_l^i)$. Where Φ_k is the negative truncated signed distance function (SDF) associated to the mesh k , $\Phi_k(\mathcal{V}) = \max(0, -SDF(\mathcal{V}))$.

$$\mathcal{L}_{col} = \sum_{k,l} \mathcal{L}_{col}^{k,l} \quad (5.5)$$

This formulation allows to handle any number of visible hands and objects in the scene.

Local contacts. Hands interact with objects by establishing surface contacts without interpenetration. We experiment with the hand-object heuristic introduced by Hasson et al. (2019). We re-purpose this loss which has been introduced in a learning framework to our optimization setup. This additional term encourages the contacts to occur at the surface of the object by penalizing the distance between hand vertices the closest object vertex for hand vertices in the object’s vicinity. We refer to the supplemental material for additional details.

The final objective \mathcal{L} is composed of a weighted sum of the previously described terms, where the weights are empirically set to balance the contributions of each error term.

$$\begin{aligned} \mathcal{L} = & \lambda_{obj} \mathcal{L}_{obj} + \lambda_{v2d} \mathcal{L}_{v2d} + \lambda_{pca} \mathcal{L}_{pca} + \lambda_{scale} \mathcal{L}_{scale} \\ & + \lambda_{smooth} \mathcal{L}_{smooth} + \lambda_{centroid} \mathcal{L}_{centroid} + \lambda_{local} \mathcal{L}_{local} + \lambda_{col} \mathcal{L}_{col} \end{aligned} \quad (5.6)$$

While Zhang et al. (2020) adapt the weights for their optimization for each object categories, we fix the weight parameters empirically and keep them constant across all experiments. We refer to the supplemental material for exact weight values and additional implementation details.

5.3 Learning from noisy data

Finally, we explore the synergies between our bottom-up fitting method and existing learnt methods. Our method can adapt to any provided model at inference time. In contrast, recent learning-based methods for hand-object pose estimation such as those introduced in Tekin et al. (2019) and in Chapter 4 typically output instance-specific poses, and require access to the inference model at train time. Here, we further improve the obtained fitted object poses using a learnt model. We perform automatic self-labelling using our fitting method on the test set from the HO-3D dataset, and train a neural network model simultaneously on the training set of HO-3D and on our noisy labels obtained on the test set. For this, we employ the architecture introduced in Chapter 4. We perform automatic filtering by removing object fits which present a high discrepancy between the final rendered object mask and the 2D segmentation mask, which is either due to an incorrect fit to the 2D evidence or an error in the predicted segmentation mask. We refer to the supplemental material for details on the exact threshold criterion. By leveraging both the ground truth training labels and noisy object estimates, our method can benefit from previous 3D labeling efforts to provide more accurate estimates for manipulated objects.

5.4 Experiments

We first define the evaluation metrics (Section 5.4.1) and the datasets (Section 5.4.2) used in our experiments. Then, we provide an ablation to measure the contribution of each of our optimization objective terms (Section 5.4.3). We investigate the sensitivity of our approach to the quality of the 2D estimates (Section 5.4.4). Next, we compare our approach to the state of the art (Section 5.4.5). Finally, we provide qualitative results for in-the-wild examples (Section 5.4.6).

5.4.1 Metrics

The structured output for hand-object reconstruction is difficult to evaluate with a single metric. We therefore rely on multiple evaluation measures.

Object metrics. We evaluate our object pose estimates by computing the average vertex distance. Common objects such as bottles and plates often present plane and revolution symmetries. To account for point matching ambiguities, we further report the standard pose estimation average closest point distance (add-s) Xiang et al. (2018).

Hand metrics. We follow the standard hand pose estimation protocols Hampali et al. (2020); Zimmermann et al. (2019) and report the procrustes-aligned hand vertex error and

F-scores. We compare the hand joint predictions using average distances after scale and translation alignment. When investigating the results of the joint fitting, we additionally report the average hand vertex distances without alignment.

Interaction metrics. *Penetration depth (mm)*: We report the maximum penetration depth between the hand and the object following previous work on hand-object interactions Brahmbhatt et al. (2020); Karunratanakul et al. (2020) as well as Chapters 3 and 4. *Contact (%)*: We also report the contact percentage following Karunratanakul et al. (2020). When ground truth contact binary labels are available, we report contact accuracy, as we further detail in the supplemental material.

5.4.2 Datasets

HO-3D Hampali et al. (2020) is the largest dataset to date to provide accurate hand-object annotations during interaction for marker-less RGB images. The users manipulate 10 objects from the YCB Çalli et al. (2015) dataset, for which the CAD models are provided. The ground-truth annotations are obtained by fitting the hand and object models to RGB-D evidence which assumes limited hand motion. We present all results on the test set, which is composed of 13 videos for a total of 11525 frames depicting single-hand object manipulations.

Core50 Lomonaco and Maltoni (2017) contains short sequences of unannotated images of hands manipulating 50 object instances from 10 everyday object categories such as cups, light bulbs and phones. We manually associate 26 objects with approximately matching 3D object models from the ShapeNet dataset Chang et al. (2015). We further annotate hands being left or right for each of the 11 video sequences available for each object, resulting in 286 video clips and 86k frames.

Epic Kitchens Damen et al. (2018) is an unscripted dataset which has been collected without imposing constraints or equipment beyond a head-mounted camera. In contrast to existing datasets for which 3D information is available, it therefore presents natural hand-object interactions. This dataset is however densely annotated with action labels which include the category of the object of interest. We focus on a subset of common object categories: cups, plates, cans, phones and bottles which are involved in a total of 3456 action video clips. For each object category, we associate an object model from the CAD ShapeNet Chang et al. (2015) database. We assume that the target manipulated object is the one with the longest track in the associated action clip.

	Hand		Object		Interaction	
	vertex mean distance (cm)↓	mepe aligned (cm) ↓	vertex mean distance (cm) ↓	add-s (cm) ↓	pen. depth (mm) ↓	contact %
indep. composition	26.2	5.2	12.1	7.7	3.2	25.8
joint fitting	8.6	5.4	8.1	3.8	2.8	77.5
w/out local interactions	8.5	5.4	8.0	3.8	2.4	72.3
w/out collision	8.9	5.4	8.0	3.8	10.2	80.5
w/out coarse interaction	17.1	5.3	8.1	3.8	1.9	59.4
w/out smoothness	11.4	5.6	12.8	8.3	3.0	79.1

Table 5.1: **Contribution of error terms:** We show benefits of the *joint* modeling for hand-object interactions by the increased reconstruction accuracy when compared to independent hand and object composition on the HO-3D Hampali et al. (2020) dataset. Our smoothness and interaction terms impose additional constraints which improve the final hand-object pose reconstructions.

5.4.3 Contribution of error terms in fitting

As explained in Section 5.2, our method introduces several error terms which determine the final reconstruction quality. We evaluate the contribution of the main error terms on the HO-3D benchmark Hampali et al. (2020) in Table 5.1. We validate that our joint reconstruction outperforms the naive composition baseline, which is obtained by separately fitting the object to the occlusion-aware object mask and the hand using the hand-specific terms \mathcal{L}_{v2d} and \mathcal{L}_{pca} . When fitted independently, the scale-depth ambiguity prevents an accurate estimate of the hand distance. As we use the ground-truth object model, the 3D object pose can be estimated without ambiguity using the camera intrinsic parameters. Joint fitting improves the 3D pose estimates using both smoothness and interaction priors. We observe that the coarse interaction prior is critical towards improving the absolute hand pose. When removing this error term, the hand pose increases two-fold from 8.9 to 17.1cm. We observe that the temporal smoothness term, while simple, provides a strong improvement to both the hand and object pose estimates. Leveraging information across neighboring video frames reduces the errors by 25% and 38% for the hand and object, respectively. While the local interaction and collision penalization terms only marginally change the hand and object reconstruction scores, their impact can be quantitatively observed in the interaction metrics. The collision penalization terms reduce the average penetration depth by a large factor (10.2mm vs 2.4mm). The local interaction term introduced in Chapter 3 reduces both the interpenetration depth and the contact percentage, which is defined as either exact surface contact or interpenetration between the hand and the object. Qualitatively, we observe that this term produces local corrections in the vicinity of estimated contact points. Including all error terms results in more plausible grasps, which we illustrate in Figure 5-2 qualitatively.

While ground truth 3D poses are hard to annotate for generic videos with hand-object manipulations, interaction metrics such as penetration depth can be directly computed

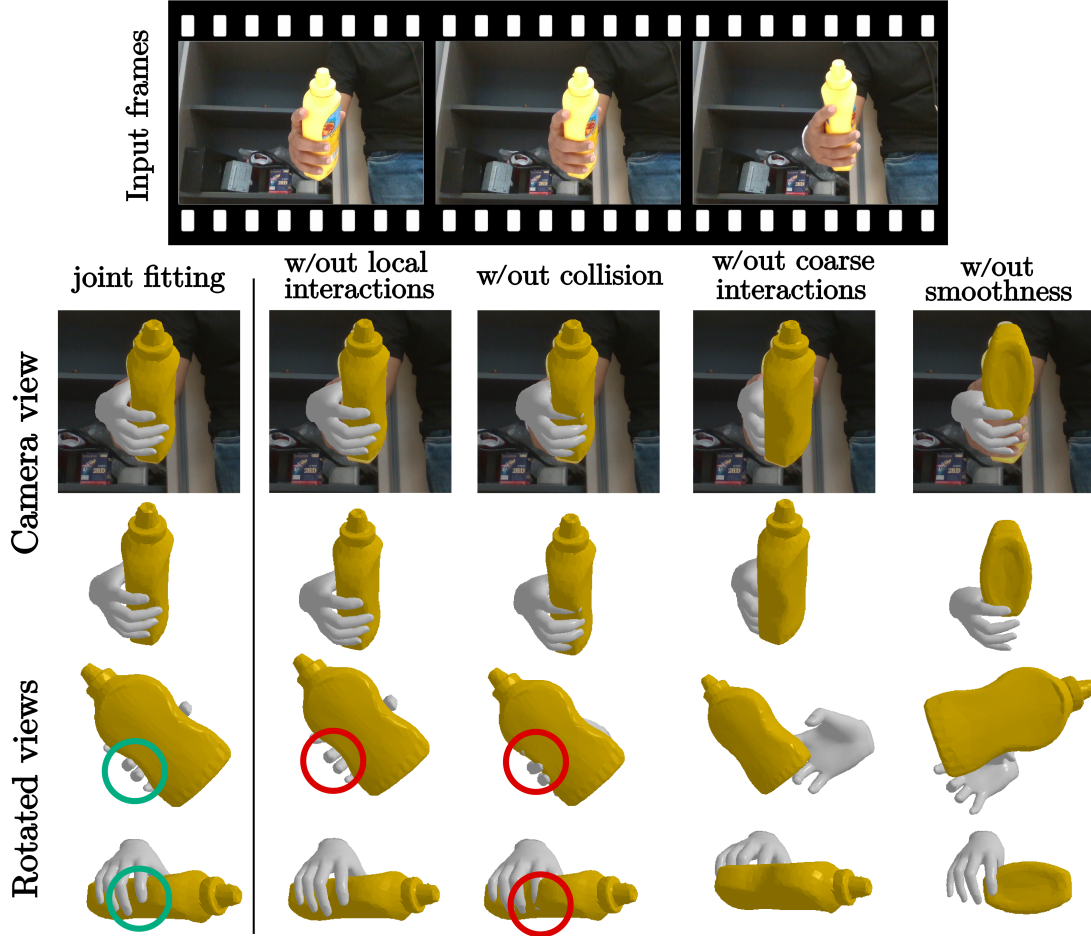


Figure 5-2: **Effect of error terms:** Qualitative analysis showing the effects of the various error terms for the hand-object reconstruction accuracy on the HO-3D dataset. We highlight visual evidence of local corrections attributed to the local interaction from Chapter 3 and collision Jiang et al. (2020) terms.

from the predicted reconstructions. As the Core50 Lomonaco and Maltoni (2017) dataset presents only videos in which the object is actively manipulated by the hand, we can additionally report contact accuracy as proxies to evaluate the quality of the reconstructed grasps. We report these two metrics on the Core50 dataset in Tables. 5.2 and confirm the benefit from our joint fitting approach. Using joint fitting allows to reconstruct configurations which are in contact for 90% of the frames, while only increasing by 0.6mm the average penetration depth.

5.4.4 Sensitivity to estimated 2D evidence

Our method makes use of generic models for detection and mask estimation and would directly benefit from more accurate detection, 2D hand pose estimation and instance seg-

Dataset	Contact Accuracy \uparrow		Pen. Depth (mm) \downarrow	
	independent	joint	independent	joint
Core50	7.3	89.5	0.6	1.2

Table 5.2: **Results on Core50:** Interaction errors for hand-object fits obtained on the Core50 dataset. We observe significantly improved contact accuracy with joint fitting over independent fits at the expense of a minor cost of a 0.6mm increase in penetration.



Figure 5-3: **Sensitivity to 2D detections:** Dependence of our 3D reconstruction on the accuracy of the 2D evidence by running our method with ground truth (GT) hand and object detections and ground truth object masks for the HO-3D dataset Hampali et al. (2020).

mentation models. The reliance of our method on 2D cues therefore allows it to benefit from additional efforts in 2D image annotation which is simpler compared to 3D annotation in practice. We investigate the dependence of our method on the quality of the available 2D evidence. To investigate the expected improvements our method could gain from stronger object detections, instead of using noisy detections, we use the hand and object ground truth bounding boxes provided for the HO-3D test set. We observe in Figure 5-3 the improvements we obtain from using the ground-truth detections. Both hands and objects benefit from more accurate detections, improving by 2cm when compared to the tracking-by-detection estimates. To investigate the errors which come from using noisy approximate instance masks, we render the object and hand ground truth masks and use them to guide our optimization. By relying on 2D information, our approach suffers from limitations such as depth ambiguities which can result from fitting to image segmentation masks. Object asymmetries which rely on color information can also be hard to resolved during fitting. We observe that using ground truth hand and object masks allows to further decrease the 3D pose errors. we note that the object error decreases to 2cm and below 1cm cm when comparing distances to closest points. When the object model is available, our joint fitting method produces highly accurate object poses in the presence of accurate 2D evidence.

Method	mesh error ↓	F-score @5mm ↑	F-score F@15mm ↑	aligned mepe ↓
Hampali 2020 Hampali et al. (2020)	1.06	0.51	0.94	3.0
Hasson 2019 Hasson et al. (2019)	1.10	0.46	0.93	3.2
Hasson 2020 Hasson et al. (2020)	1.14	0.42	0.93	3.7
Joint fitting	1.47	0.39	0.88	5.5
Joint fitting (GT tracks)	1.50	0.44	0.93	4.1

Table 5.3: **State-of-the-art-comparison:** We compare the hand performance of the single-view baseline from Chapter 4 to previously reported methods on hand metrics. Note that the reported results for Hampali et al. (2020) and Chapter 3 are for methods which output hand meshes only, Chapter 4 and the method presented in this chapter predict the hand-object meshes jointly. All methods are trained only on the real images from the HO-3D training split and evaluated on the official test split through an online submission ¹.

5.4.5 State-of-the-art comparison

Recent efforts for joint hand-object pose estimation in camera space Tekin et al. (2019) as well as Chapters 3 and 4 have focused on direct bottom-up regression of 3D poses. We compare the performance of our fitting approach to our recent learning-based method for joint hand-object reconstruction Chapter 4. In Table 5.3, we note that the joint learnt hand-object model we introduced in Chapter 4 has a slightly lower hand accuracy when compared with the current state-of-the art Hampali et al. (2020). However, to the best of our knowledge, the work presented in Chapter 4 was the only published work to have provided a trained model for joint hand and object pose estimation trained on the HO-3D Hampali et al. (2020) dataset at this time.

We compare our method to the learnt baseline when using (i) automatically tracked and (ii) ground-truth bounding boxes. While the learnt baseline produces more accurate hand predictions, its object predictions are *instance specific*. As a direct consequence, the method does not generalize to new objects at test time. In contrast, our generic fitting method performs equally well across the seen and unseen objects of the HO-3D test split, see Table 5.4.

We investigate the complementarity of the learnt baseline and our fitting approach through a test-time training strategy. We train a joint hand-object pose estimation model with the architecture described in Section 4.2 on both labelled examples from the official train set and a subset of automatically selected noisy HO-3D test set images. We obtain improved estimates of object poses on the test set when compared to both the learnt model or the fitting method applied independently. In particular, the error on unseen object decreases

¹<https://competitions.codalab.org/competitions/22485>

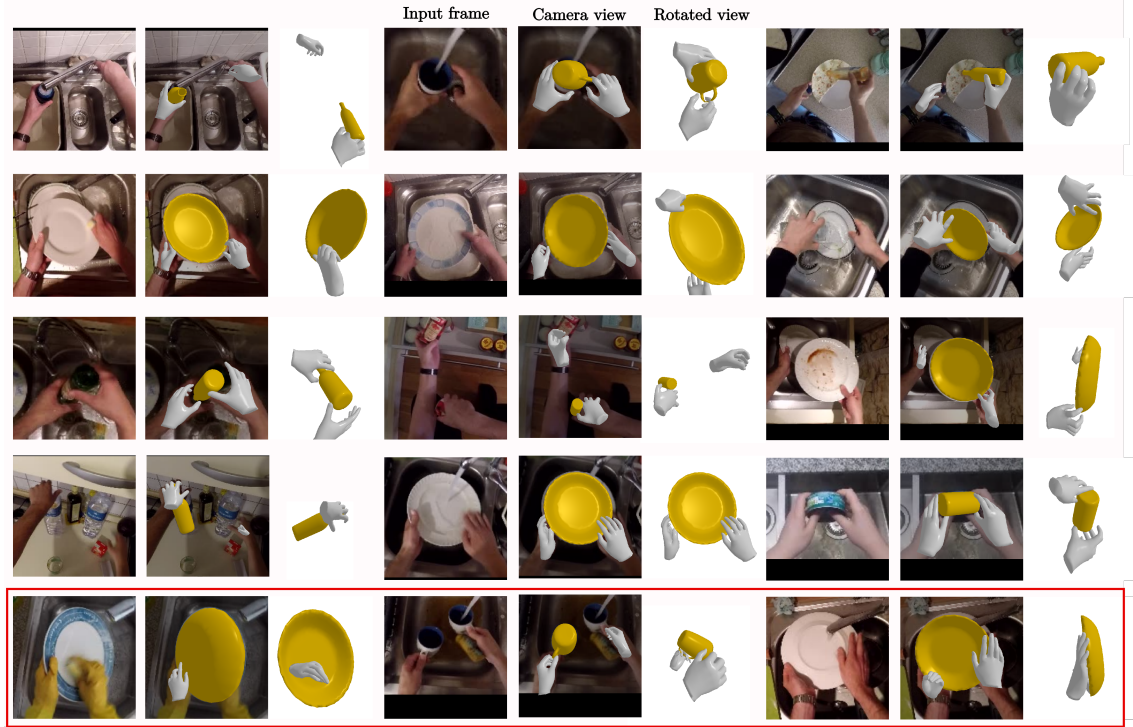


Figure 5-4: **In-the-wild reconstructions:** Our results on natural hand-object manipulations of the Epic-Kitchens dataset Damen et al. (2018). We present several success and failures of our method on the challenging Epic-Kitchens dataset. We highlight typical failure modes for our method, in particular, object orientation errors resulting from depth ambiguity. We observe that our fitting method recovers plausible interactions across different object categories and hand-object configurations.

	Object				Hand	
	vertex dist (cm) ↓		add-s (cm) ↓		mepe ↓	aligned mepe ↓
	Seen	Unseen	Seen	Unseen		
Ours	8.0	8.1	4.0	3.3	8.6	5.4
Hasson et al. (2020)	6.7	10.7	2.2	3.6	5.5	3.7
Ours + test-time tr.	5.5	5.4	1.9	1.7	6.2	4.4

Table 5.4: **Unseen objects:** Vertex errors (cm) for estimated hand and object meshes. Compared to Chapter 4, our method performs similarly across seen and *unseen* objects and sees further benefits from test-time training.



Figure 5-5: **Comparison with Chapter 3:** Qualitative comparison of our fits to Chapter’s 3 ObMan-trained model estimations on the Core50 dataset. While our model requires an approximate mesh to be provided, it generalizes to objects of arbitrary topology.

by 2.7cm compared to the original unfiltered fitted poses.

5.4.6 In-the-wild 3D hand-object pose estimation

We test the limits of our approach and showcase the strength of our method by comparing to the model trained for joint hand-object reconstruction introduced in Chapter 3. Their model estimates the shape of the object by deforming a sphere and therefore does not depend on known object models. However, given this object topology restriction, this method has limited expressivity. It can only capture a subset of all object shapes which excludes everyday objects such as mugs or cups, see Fig. 5-5. In comparison, while our method makes stronger assumptions by relying on an approximate object model, it is applicable to any everyday objects for which an approximate mesh can be retrieved without further limitations. Additionally, while the manipulation reconstruction from Chapter 3 estimates the grasp relative to the root joint of the hand, our method outputs image-aligned predictions.

We further show that our method can be applied to the challenging Epic-Kitchens dataset Damen et al. (2018) which presents natural manipulation of common objects, see Fig. 5-4. Note that our objective (5.6) is not restricted to a single hand-object pair and naturally generalizes to multiple hands and objects. To handle scenes with two hands in Damen et al. (2018) we optimize (5.6) with pairwise losses defined for both detected hands and the detected object. We show results of two-hand manipulations which represent the majority of examples in the target dataset. While we observe cases of depth ambiguity, especially with almost planar objects such as plates, we show that our method can recover plausible reconstructions across a variety of object categories and hand poses.

5.5 Conclusions

We presented an approach for fitting 3D hand-object configurations to monocular RGB videos. Our method builds on estimates obtained with neural network models trained with full supervision. Due to lack of supervision at similar scale for 3D, we opt for a fitting-based approach. We demonstrate that our method allows to reconstruct hand-object interactions on several datasets to which current learning-based methods can not be applied for lack of training data. While our proposed method is less competitive when compared to learning methods trained on the HO-3D train set, it is applicable to domains for which training data is not available, such as the Epic-Kitchens dataset Damen et al. (2018).

Our method assumes an exact or approximate object model to be known for each video clip. Additionally reconstructing or retrieving the target object in each sequence would allow to further progress towards fully automatic joint modeling from RGB inputs.

Chapter 6

Discussion

In this chapter, we summarize the contributions of this thesis (Section 6.1) and outline challenges and directions for future work (Section 6.2).

6.1 Summary of contributions

This thesis has focused on the problem of modeling hand-object interactions from RGB images. Our contributions are the following:

- In Chapter 3 we generate a large-scale synthetic dataset of hand-object interactions with rich ground truth annotations. We demonstrate that such data can be used to train a CNN which regresses the object shape and hand pose from a single RGB frame. As our reconstruction model is end-to-end differentiable, we propose to use a novel contact loss to encourage plausible grasps at train time and qualitatively improve the reconstructions.
- In Chapter 4, we propose to leverage the temporal context of sparsely annotated frames. We show that we can use a simple loss based on photometric consistency to provide additional supervision with weak supervision from neighboring frames.
- In Chapter 5 we propose to use a different source of supervision: noisy predictions from learnt models for hand and object detection, instance segmentation and isolated hand pose estimation. We used joint fitting, optimizing hand and object pose for a known CAD model. We demonstrate that this procedure allows to recover plausible hand-object reconstructions, providing noisy annotations for short RGB video clips.

6.2 Perspectives

As we have emphasized throughout this thesis, annotated data is one of the most critical resources required to develop robust reconstruction methods under partial viewing conditions. We summarize the limitations of current datasets in Section 6.2.1. We then discuss two orthogonal strategies which could improve in-the-wild hand-object reconstruction. We first argue in favor of improving annotation tools for interacting hands and objects in Section 6.2.2 and discuss learning from noisy data as an alternative strategy which could circumvent the need for manual 3D annotation in Section 6.2.3.

The ability to accurately estimate how the hand interacts with objects in everyday scenes would enable interesting future work. We present two direct application which could be explored. We argue in favor of modeling object state changes from videos in Section 6.2.4 and object’s visual affordances discovery in Section 6.2.5

6.2.1 Importance and limitations of existing datasets.

Our work presented in Chapters 3, 4 and 5 was enabled by the release of the FPHAB dataset by Garcia-Hernando et al. (2018) and, one year later, the release of the HO-3D dataset from Hampali et al. (2019). These datasets provide benchmarks on which learning methods can be compared. However, models trained on these datasets typically do not generalize to new domains because they regress instance-specific object poses or because they overfit to limited training data. More recent datasets such as DexYCB Chao et al. (2021), ContactPose Brahmabhatt et al. (2020) and H2O Kwon et al. (2021), though one or two orders of magnitude larger in terms of annotated frames, present the same limitations.

Two important limitations currently prevent in-the-wild generalization:

- A lack of diverse and sufficient annotated data, which could be used to train model which would generalize to unconstrained videos . such as the ones available on the web
- A focus on methods which target the limited existing datasets. Existing methods often do not explicitly target unknown object instances or cross-domain generalization.

6.2.2 Hybrid annotation methods.

It is still an open problem how to increase the diversity and quantity of annotated hand-object interaction images. Improving the ease and speed of annotation for RGB images would allow to curate larger datasets with increased diversity. Developing efficient annotation tools has been an important focus in object segmentation and 3D shape estimation,

and has arguably proven instrumental for providing robust models which generalize in-the-wild (Castrejón et al. (2017); Sun et al. (2018)). A similar effort for hand-object annotations would most likely spur important progress in manipulation modeling.

Current methods, such as the ones we presented in Chapter 3 and 5, which reconstruct the 3D from images could be used to speed up annotations. For instance, results from fitting could be filtered and refined by a human annotator using interactive interfaces where users could select and correct annotation proposals. An investment in improving annotation interfaces to lower the tediousness of annotation for arbitrary objects and hands in challenging viewing conditions would allow to develop stronger models, and bias research efforts towards methods which generalize in the wild.

6.2.3 Learning from noisy annotations.

In Chapter 5, we proposed a method to recover noisy annotations for RGB frames by fitting to evidence collected for hands and objects separately. Such noisy annotations would prove of high value if we can demonstrate that they can be used to train models which generalize out of domain and outperform fitting approaches such as the one we presented in Chapter 5 and the work of Cao et al. (2020b). Noisy annotations have proven useful for training learning-based methods for body (Arnab et al. (2019)) and hand pose (Kulon et al. (2020)) estimation. Using noisy annotations for training provides an opportunity to leverage increased and more diverse source of data, addressing the stringent limitations of current datasets. Learning from noisy data provides an orthogonal direction to collecting more annotations. In contrast to hybrid annotation efforts, which would still be limited by a manual bottleneck, methods trained with automatic noisy supervision could directly benefit from the large-scale availability of unannotated data. Such approaches however would need to explicitly account for the noise in the data, for instance by automatically detecting and filtering poor reconstructions or by designing methods intrinsically robust to noise.

6.2.4 Modeling object state changes

In the previous sections, implicitly or explicitly, we have focused mostly on rigid objects, sometimes extending our scope to articulated ones. However, a large fraction of human daily actions involve more drastic changes in objects' states. Think of the last times for instance when you prepared a meal. Most likely, it involved changing the state of at least several ingredients and other objects, potentially cutting, mixing, bending or ripping them, as prominently featured in the Epic-Kitchens dataset. The community currently lacks datasets which model 3D state changes for objects during manipulation are limited. Many chal-



Figure 6-1: Recent methods which model hand-object interactions can only model a restricted subset of objects, and typically do not model object state changes. Addressing these dynamic scenes which are beyond the scope of existing reconstruction methods requires carefully designing appropriate 3D scene representations.

Challenges arise from modeling the temporal evolution of objects. Some changes such as object deformation can be modeled using existing techniques and representation, for instance by tracking points on the surface of an object mesh through time. However, many basic actions such as cutting an apple in several parts or "Tearing something into two pieces", a category from the Something Something dataset (Corney et al. (2005)), would raise non-trivial modeling questions. These examples highlight the limits of simplified assumptions we made in the introduction: the simple agent/active object/passive environment factorization and a single active object (though Chapter 5.2.3 could be extended to include more objects). Modeling more complex scenes over time requires exploring appropriate representations. A good such representation would capture the scene's 3D evolution in a space which would facilitate the interpretation about the resulting action. We believe that exploring 3D representations for this purpose would be of high interest and suggest starting by modeling simple actions which result in object state changes. An interesting first step in that direction could target compositional changes between rigid object parts such as opening and closing a bottle, in which the cap and the body of the object go from two separate entities to a single object composed of the two securely joined parts. A second more challenging task could focus on actions such as cutting or ripping objects into several parts, as we illustrate in Figure 6-1, which raises the subject of properly modelling a changing number of parts of unknown shapes.

6.2.5 Discovering statistical object affordances

Improved methods for in-the-wild hand-object annotation or estimation could extract diverse plausible hand-object interactions from real-world examples. Modeling affordances, "action possibilities in the environment in the relation to the action capabilities of the actor" (Gibson (1966)), is an interesting application. In the case of object manipulations, we are interested in modeling and possibly sampling actions for a given agent, object and environment. In a 3D scene, we distinguish between physical affordances, the set of all state changes which can be imposed by the agent to a target object, and statistical affordances,

the distribution of manipulations which are likely to be performed by humans.

Several work have predicted affordances as labelled heatmaps in pixel space for scenes (Defferrard et al. (2016); Fouhey (2015)) or objects (Fang et al. (2018); Nagarajan et al. (2019)). Recent work have turned to predicting possible hand poses or human motions to capture scene affordances. Corona et al. (2020) predict possible hand grasps for objects observed in RGB images while Cao et al. (2020a) sample target locations in a room and trajectories conditioned on these destinations to predict valid human displacements. A similar formulation could be applied for hand-object interactions, where affordances could be expressed as the set of valid trajectories or motions which the hand can execute given the current state of the environment. Modeling affordances as 3D changes in time could provide a useful intermediate representation for later transfer to automatically controlled agents. Formulating affordances as predicting candidate motions would provide interpretable answers to the questions "What can I do in this scene" and "How can I achieve a specific goal".

Learning affordances from diverse demonstrations featured for instance in the Something-Something (Goyal et al. (2017)) HowTo100M (Miech et al. (2019)) and Epic-Kitchens (Damen et al. (2018)) datasets could improve object manipulation planning. Affordances modeled as 3D motions or trajectories could be directly retargeted to simulated environments, as we schematically illustrate in Figure 6-2. Given a scene and an intention, formulated as an action label for instance, generated motion proposals could guide an agent in its exploration of a simulated environment. Very recently, Mandikal and Grauman (2021) predict affordances as heatmaps from synthetic images and use this guidance to grasp objects in simulation. We believe an interesting direction would be to target more challenging 3D environments such as SAPIEN introduced in Xiang et al. (2020) or iGibson described in Shen et al. (2020). Both SAPIEN and iGibson propose to integrate composite objects in simulated scenes, affording actions beyond grasping and displacing such as opening drawers or laptops.

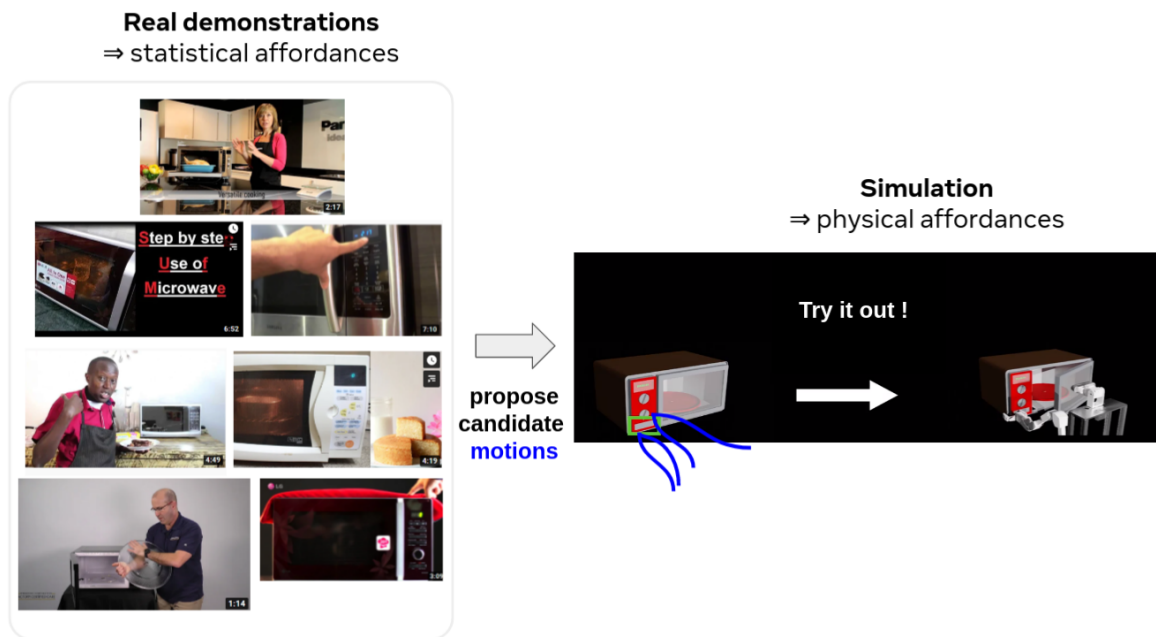


Figure 6-2: Reconstructing hand-object interactions from a large number of demonstration videos could allow to capture statistical object affordances. Real demonstrations could be used to extract plausible agent motions which could guide exploration in simulated environments such as SAPIEN Xiang et al. (2020) or iGibson Shen et al. (2020). The figure is composed using illustrations from the SAPIEN dataset (Xiang et al. (2020)) and frames from YouTube videos describing microwave usage.

RÉSUMÉ

Modéliser la manipulation d'objets est essentiel à la compréhension des interactions entre l'homme et son environnement. Les efforts pour reconstituer l'information 3D à partir d'images en couleur se sont récemment orientés vers les méthodes d'apprentissage supervisées, qui requièrent de nombreux exemples annotés pour l'entraînement. Cependant, la collecte d'annotations précises pour des images de manipulation est laborieuse, coûteuse, et propice aux erreurs.

Cette thèse propose plusieurs stratégies pour pallier ces difficultés en utilisant des données synthétiques, le contexte temporel, ou des prédictions bruitées d'autres modèles.

MOTS CLÉS

Vision artificielle, Reconstruction 3D, Modélisation de l'activité humaine, Analyse de la manipulation d'objets, Réseaux de neurones convolutionnels, Apprentissage de la représentation.

ABSTRACT

Modeling hand-object manipulations is essential for understanding how humans interact with their environment. Recent efforts to recover 3D information from RGB images have been directed towards fully-supervised methods which require large amounts of labeled training samples. However, collecting 3D ground-truth data for hand-object interactions is costly, tedious, and error-prone.

In this thesis, we propose several contributions to overcome this challenge using synthetic data, temporal context and fitting to pseudo-labels from learnt models.

KEYWORDS

Computer vision, 3D reconstruction, Human activity modeling, Object manipulation understanding, Convolutional neural networks, Representation learning.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Akenine-Möller, T. and Trumbore, B. (1997). Fast, minimum sotrage ray-triangle intersection. *Journal of graphics tools*.

Anguelov, D. (2005). *Learning Models of Shape from 3D Range Data*. PhD thesis, Stanford.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: shape completion and animation of people. *ACM transactions on graphics*.

Argyros, A. and Lourakis, M. (2004). Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*.

Arnab, A., Doersch, C., and Zisserman, A. (2019). Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*.

Athitsos, V. and Sclaroff, S. (2003). Estimating 3d hand pose from a cluttered image. In *CVPR*.

Aubry, M., Maturana, D., Efros, A., Russell, B., and Sivic, J. (2014). Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*.

Baek, S., Kim, K. I., and Kim, T.-K. (2019). Pushing the envelope for rgb-based dense 3D hand pose estimation via neural rendering. In *CVPR*.

- Balasubramanian, R., Xu, L., Brook, P. D., Smith, J. R., and Matsuoka, Y. (2012). Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Trans. on Robotics*.
- Ballan, L., Taneja, A., Gall, J., Van Gool, L., and Pollefeys, M. (2012). Motion capture of hands in action using discriminative salient points. In *ECCV*.
- Bansal, A., Russell, B., and Gupta, A. (2016). Marr Revisited: 2D-3D model alignment via surface normal prediction. In *CVPR*.
- Barr, A. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*.
- Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *IJCV*.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *ICCV*.
- Bertasius, G., Feichtenhofer, C., Tran, D., Shi, J., and Torresani, L. (2019). Learning temporal pose estimation from sparsely-labeled videos. In *NeurIPS*.
- Besl, P. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*.
- Binford, T. (1971). Visual perception by computer. In *IEEE Conference on Systems and Control*.
- Blender Online Community (2018). Blender - a 3D modelling and rendering package.
- Bogo, F., Kanazawa, A., Lassner, C., Gehler, V. P., 0002, R. J., and Black, J. M. (2016). Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. *ECCV*.
- Bohg, J., Morales, A., Asfour, T., and Kragic, D. (2014). Data-driven grasp synthesis - A survey. *IEEE Trans. Robotics*.
- Boukhayma, A., Bem, R. d., and Torr, P. H. (2019). 3d hand shape and pose from images in the wild. In *CVPR*.

- Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C. (2014). Learning 6d object pose estimation using 3d object coordinates. In *ECCV*.
- Brachmann, E., Michel, F., Krull, A., Yang, M. Y., Gumhold, S., and Rother, c. (2016). Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*.
- Brahmbhatt, S., Handa, A., Hays, J., and Fox, D. (2019). ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact. In *IROS*.
- Brahmbhatt, S., Tang, C., Twigg, C. D., Kemp, C. C., and Hays, J. (2020). ContactPose: A dataset of grasps with object contact and hand pose. In *ECCV*.
- Breiman, L. (2001). Random forests. *Machine Learning*.
- Brickwedde, F., Abraham, S., and Mester, R. (2019). Mono-sf: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes. In *ICCV*.
- Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *ECCV*.
- Cai, M., Kitani, K. M., and Sato, Y. (2015). A scalable approach for understanding the visual structures of hand grasps. In *ICRA*.
- Cai, Y., Ge, L., Cai, J., and Yuan, J. (2018). Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*.
- Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.-J., Yuan, J., and Thalmann-Magnenat, N. (2019). Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In *ICCV*.
- Calinon, S., Guenter, F., and Billard, A. (2006). On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics*.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Cao, Z., Gao, H., Mangalam, K., Cai, Q., Vo, M., and Malik, J. (2020a). Long-term human motion prediction with scene context. In *Eur. Conf. Comput. Vis.*
- Cao, Z., Radosavovic, I., Kanazawa, A., and Malik, J. (2020b). Reconstructing hand-object interactions in the wild. *arXiv preprint arXiv:2012.09856*.

- Carnegie Mellon University (2001). Carnegie-Mellon Mocap Database. <http://mocap.cs.cmu.edu/>.
- Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.
- Castrejón, L., Kundu, K., Urtasun, R., and Fidler, S. (2017). Annotating object instances with a polygon-rnn. In *CVPR*.
- Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*.
- Chao, Y.-W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y., Wyk, K. V., Iqbal, U., Birchfield, S., Kautz, J., and Fox, D. (2021). Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*.
- Chen, W., Gao, J., Ling, H., Smith, E., Lehtinen, J., Jacobson, A., and Fidler, S. (2019). Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*.
- Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2015). Synthesizing training images for boosting human 3d pose estimation. In *3DV*.
- Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *CVPR*.
- Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*.
- Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*.
- Connolly, K. and Dalgleish, M. (1989). The emergence of a tool-using skill in infancy. *Developmental Psychology*.
- Corney, J., Hayes, C., Sundararajan, V., and Wright, P. K. (2005). The CAD/CAM interface: A 25-year retrospective. *Journal of Computing and Information Science in Engineering*.

- Corniani, G. and Saal, H. P. (2020). Tactile innervation densities across the whole body. *Journal of Neurophysiology*.
- Corona, E., Pumarola, A., Alenya, G., Moreno-Noguer, F., and Rogez, G. (2020). Gan-Hand: Predicting human grasp affordances in multi-object scenes. In *CVPR*.
- Coumans, E. (2013). Bullet real-time physics simulation.
- Coumans, E. and Bai, Y. (2016–2019). Pybullet, a python module for physics simulation for games, robotics and machine learning.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.
- Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*.
- de La Gorce, M., Fleet, D. J., and Paragios, N. (2011). Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*.
- Del Pero, L., Bowdish, J., Kermgard, B., Hartley, E., and Barnard, K. (2013). Understanding bayesian rooms using composite 3d object models. In *CVPR*.
- Delingette, H. (1994). Simplex meshes: a general representation for 3d shape reconstruction. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.
- Deprelle, T., Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. (2019). Learning elementary structures for 3d shape generation and matching. In *Neurips*.
- Dhome, M., Richetin, M., Lapresté, J.-T., and Rives, G. (1989). Determination of the attitude of 3-d objects from a single perspective view. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dictionnaire de l'Académie Française (2021). Objet.

Dixon, S. D. (2006). *Encounters with children : pediatric behavior and development*. Mosby Elsevier, Place of publication not identified.

Doosti, B., Naha, S., Mirbagheri, M., and Crandall, D. (2020). Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*.

Douvantzis, P., Oikonomidis, I., Kyriazis, N., and Argyros, A. (2013). Dimensionality reduction for efficient single frame hand pose estimation. In *ICVS*.

Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*.

Everingham, M., Eslami, A., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *IJCV*.

Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*.

Fang, K., Wu, T.-L., Yang, D., Savarese, S., and Lim, J. J. (2018). Demo2vec: Reasoning object affordances from online videos. In *CVPR*.

Feix, T., Pawlik, R., Schmiedmayer, H.-B., Romero, J., and Kragic, D. (2009). A comprehensive grasp taxonomy. In *RSS Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*.

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

Ferrari, C. and Canny, J. F. (1992). Planning optimal grasps. In *ICRA*.

Fidler, S., Dickinson, S., and Urtasun, R. (2012). 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NeurIPS*.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.

Fouhey, David F., W. X. G. A. (2015). In defense of the direct perception of affordances. *arXiv preprint arXiv:1505.01085*.

- Gall, J., Rosenhahn, B., and Seidel, H. (2006). An introduction to interacting simulated annealing. In *Human Motion, Understanding, Modelling, Capture, and Animation*, Computational Imaging and Vision.
- Gall, J., Yao, A., Razavi, N., Van Gool, L., and Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018). First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*.
- Ge, L., Liang, H., Yuan, J., and Thalmann, D. (2017). 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *CVPR*.
- Ge, L., Ren, Z., and Yuan, J. (2018). Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Geman, S. and McClure, D. E. (1987). Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*.
- Gibson, J. J. (1966). *The senses considered as perceptual systems*. Houghton Mifflin.
- Gilbert, E., Johnson, D., and Keerth, S. (1988). A fast procedure for computing the distance between complex objects in three-dimensional space. *Journal of Robotics and Automation*.
- Girshick, R. (2015). Fast r-cnn. In *ICCV*.
- Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *CVPR*.
- Goel, S., Kanazawa, A., , and Malik, J. (2020). Shape and viewpoints without keypoints. In *ECCV*.
- Goldfeder, C., Ciocarlie, M. T., Dang, H., and Allen, P. K. (2009). The Columbia grasp database. In *ICRA*.
- Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thureau, C., Bax, I., and Memisevic, R. (2017). The "something something" video database for learning and evaluating visual common sense. In *ICCV*.
- Grabner, A., Roth, P. M., and Lepetit, V. (2018). 3d pose estimation and 3d model retrieval for objects in the wild. In *CVPR*.

- Grabner, A., Roth, P. M., and Lepetit, V. (2019). Location field descriptors: Single image 3d model retrieval in the wild. In *3DV*.
- Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. (2018a). 3D-CODED : 3D correspondences by deep deformation. In *ECCV*.
- Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. (2018b). AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR*.
- Guler, R. A., Neverova, N., and Kokkinos, I. (2018). DensePose: Dense human pose estimation in the wild. In *CVPR*.
- Gupta, A., Efros, A., and Hebert, M. (2010). Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*.
- Guzman, A. (1968). *Computer Recognition of Three-Dimensional Objects in a Visual Scene*. PhD thesis, Massachusetts Institute of Technology.
- Hamer, H., Gall, J., Weise, T., and Van Gool, L. (2010). An object-dependent hand pose prior from sparse training data. In *CVPR*.
- Hamer, H., Schindler, K., Koller-Meier, E., and Van Gool, L. (2009). Tracking a hand manipulating an object. In *ICCV*.
- Hampali, S., Oberweger, M., Rad, M., and Lepetit, V. (2019). Ho-3d: A multi-user, multi-object dataset for joint 3d hand-object pose estimation.
- Hampali, S., Rad, M., Oberweger, M., and Lepetit, V. (2020). Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*.
- Hanbyul Joo, Hao Liu, L. T. L. G. B. N. I. M. T. K. S. N. and Sheikh, Y. (2015). Panoptic studio: A massively multiview system for social motion capture.
- Hassan, M., Choutas, V., Tzionas, D., and Black, M. J. (2019). Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*.
- Hasson, Y., Tekin, B., Bogu, F., Laptev, I., Pollefeys, M., and Schmid, C. (2020). Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*.
- Hasson, Y., Varol, G., , Laptev, I., and Schmid, C. (2021). Towards reconstructing unconstrained hand-object interactions. *arXiv preprint*.

- Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In *CVPR*.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.
- Heap, T. and Hogg, D. (1996a). 3d deformable hand models. In *IEEE International Conference on Automatic Face and Gesture Recognition*.
- Heap, T. and Hogg, D. (1996b). Towards 3D hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*.
- Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., and Lepetit, V. (2012a). Gradient response maps for real-time detection of texture-less objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., and Lepetit, V. (2011). Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*.
- Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., and Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects. In *CVPR*.
- Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2012b). Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*.
- Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., and Zabulis, X. (2017). T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *WACV*.
- Hodan, T., Sundermeyer, M., Drost, B., Labbé, Y., Brachmann, E., Michel, F., Rother, C., and Matas, J. (2020). BOP challenge 2020 on 6d object localization. In Bartoli, A. and Fusiello, A., editors, *ECCV Workshops*.
- Hodaň, T., Vineet, V., Gal, R., Shalev, E., Hanzelka, J., Connell, T., Urbina, P., Sinha, S., and Guenter, B. (2019). Photorealistic image synthesis for object instance detection. *ICIP*.
- Holzer, S., Hinterstoisser, S., Ilic, S., and Navab, N. (2009). Distance transform templates for object detection and pose estimation. In *CVPR*.

- Hossain, M. R. I. and Little, J. J. (2018). Exploiting temporal information for 3d human pose estimation. In *ECCV*.
- Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*.
- Hur, J. and Roth, S. (2017). Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *ICCV*.
- Hutmacher, F. (2019). Why is there so much more research on vision than on any other sensory modality? *Frontiers in psychology*.
- Huttenlocher, D. P., Klanderman, G., and Rucklidge, W. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Huttenlocher, D. P. and Ullman, S. (2020). Object recognition using alignment. In *ICCV*.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.
- Ingram, J., Körding, K., Howard, I., and Wolpert, D. (2008). The statistics of natural hand movements. *Experimental brain research*.
- Internet World Stats (2020). World internet usage. <https://www.internetworldstats.com/stats.htm>.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*.
- Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Iqbal, U., Molchanov, P., Breuel, T., Gall, J., and Kautz, J. (2018). Hand pose estimation via latent 2.5d heatmap regression. In *European Conference on Computer Vision (ECCV)*.
- Jacobson, A., Deng, Z., Kavan, L., and Lewis, J. P. (2014). Skinning: Real-time shape deformation. In *SIGGRAPH 2014 Courses*.
- Jarque-Bou, N., Scano, A., Atzori, M., and Müller, H. (2019). Kinematic synergies of hand grasps: a comprehensive study on a large publicly available dataset. *Journal of NeuroEngineering and Rehabilitation*.

- Jian, B. and Vemuri, B. C. (2005). A robust algorithm for point set registration using mixture of Gaussians. In *ICCV*.
- Jiang, H., Liu, S., Wang, J., and Wang, X. (2021). Hand-object contact consistency reasoning for human grasps generation. *arXiv preprint arXiv:2104.03304*.
- Jiang, W., Kolotouros, N., Pavlakos, G., Zhou, X., and Daniilidis, K. (2020). Coherent reconstruction of multiple humans from a single image. In *CVPR*.
- Johansson, R. S., Westling, G., Bäckström, A., and Flanagan, J. R. (2011). Eye–hand coordination in object manipulation. *Journal of Neuroscience*.
- Joo, H., Neverova, N., and Vedaldi, A. (2020). Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*.
- Jud, L., Fotouhi, J., Andronic, O., Aichmair, A., Osgood, G., Navab, N., and Farshad, M. (2021). Applicability of augmented reality in orthopedic surgery – a systematic review. *BMC Musculoskeletal Disorders*.
- Kanade, T. (1981). Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*.
- Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018a). End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. (2018b). Learning category-specific mesh reconstruction from image collections. In *ECCV*.
- Kanazawa, A., Zhang, J. Y., Felsen, P., and Malik, J. (2019). Learning 3D human dynamics from video. In *CVPR*.
- Kar, A., Tulsiani, S., Carreira, J., and Malik, J. (2015). Category-specific object reconstruction from a single image. In *CVPR*.
- Karras, T. (2012). Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *SIGGRAPH*.
- Karunratanakul, K., Yang, J., Zhang, Y., Black, M., Muandet, K., and Tang, S. (2020). Grasping field: Learning implicit representations for human grasps. In *3DV*.
- Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3D mesh renderer. In *CVPR*.

- Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics*.
- Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Kendall, M. and Gibbons, J. D. (1980). *Multivariate Analysis*. Charles Griffin and Company Ltd.
- Keskin, C., Kırac, F., Kara, Y., and Akarun, L. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*.
- Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., and Fitzgibbon, A. (2015). Learning an efficient model of hand shape variation from depth images. In *CVPR*.
- Khor, W., Baker, B., Amin, K., Chan, A., Patel, K., and Wong, J. (2016). Augmented and virtual reality in surgery-the digital surgical environment: Applications, limitations and legal pitfalls. *Annals of Translational Medicine*.
- Kinect (2008). <https://en.wikipedia.org/wiki/Kinect>.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.
- Kirillov, A., Wu, Y., He, K., and Girshick, R. (2019). PointRend: Image segmentation as rendering. *ArXiv:1912.08193*.
- Kjellström, H., Romero, J., and Kragic, D. (2008). Visual recognition of grasps for human-to-robot mapping. In *IROS*.
- Kokic, M., Kragic, D., and Bohg, J. (2019). Learning to estimate pose and shape of hand-held objects from RGB images. In *IROS*.
- Kovoor, J., Gupta, A., and Gladman, M. (2021). Validity and effectiveness of augmented reality in surgical education: A systematic review. *Surgery*.
- Kry, P. and Pai, D. (2006). Interaction capture and synthesis. In *SIGGRAPH*.
- Kulon, D., Guler, R. A., Kokkinos, I., Bronstein, M. M., and Zafeiriou, S. (2020). Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*.

- Kulon, D., Wang, H., Güler, R. A., Bronstein, M. M., and Zafeiriou, S. (2019). Single image 3d hand reconstruction with mesh convolutions. In *BMVC*.
- Kwon, T., Tekin, B., Stuhmer, J., Bogó, F., and Pollefeys, M. (2021). H2o: Two hands manipulating objects for first person interaction recognition. *arXiv preprint arXiv:2104.11181*.
- Kyriazis, N. and Argyros, A. A. (2013). Physically plausible 3d scene tracking: the single actor hypothesis. In *CVPR*.
- Kyriazis, N. and Argyros, A. A. (2014). Scalable 3d tracking of multiple interacting objects. In *CVPR*.
- Labbe, R. (2014). Kalman and bayesian filters in python. <https://github.com/rllabbe/Kalman-and-Bayesian-Filters-in-Python>.
- Labbé, Y., Carpentier, J., Aubry, M., and Sivic, J. (2020). CosyPose: Consistent multi-view multi-object 6D pose estimation. In *ECCV*.
- Labbé, Y., Carpentier, J., Aubry, M., and Sivic, J. (2021). Single-view robot pose and joint angle estimation via render & compare. In *CVPR*.
- Laine, S. and Karras, T. (2011). Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*.
- Lambrecht, J. and Kästner, L. (2019). Towards the usage of synthetic data for marker-less pose estimation of articulated robots in rgb images. In *ICAR*.
- Laptev, I. (2013). Modeling and visual recognition of human actions and interactions. Habilitation à diriger des recherches en mathématiques et en informatique.
- Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. In *The International Journal of Robotics Research*.
- Lepetit, V. (2020). Recent advances in 3d object and hand pose estimation.
- Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnnp: An accurate o(n) solution to the pnp problem. *IJCV*.
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*.
- Lewis, J. P., Cordner, M., and Fong, N. (2000). Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*.

- Li, Y., Wang, G., Ji, X., Xiang, Y., and Fox, D. (2018). Deepim: Deep iterative matching for 6d pose estimation. In *ECCV*.
- Lim, J. J., Pirsiavash, H., and Torralba, A. (2013). Parsing ikea objects: Fine pose estimation. In *ICCV*.
- Lin, J., Wu, Y., and Huang, T. S. (2000). Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion (HUMO'00)*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, L. (2014). Microsoft coco: Common objects in context. In *ECCV*.
- Little, J. J. and Verri, A. (1989). Analysis of differential and matching methods for optical flow. In *Proceedings Workshop on Visual Motion*.
- Liu, S., Li, T., Chen, W., and Li, H. (2019). Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *ICCV*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *ECCV*.
- Loing, V., Marlet, R., and Aubry, M. (2018). Virtual training for a real application: Accurate object-robot relative localization without calibration. *IJCV*.
- Lomonaco, V. and Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*. PMLR.
- Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*.
- Loper, M. M. and Black, M. J. (2014). OpenDR: An approximate differentiable renderer. In *ECCV*.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes A high resolution 3d surface construction algorithm. In *SIGGRAPH*.
- Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*.
- Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*.

- Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., and Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics.
- Malik, J., Elhayek, A., Nunnari, F., Varanasi, K., Tamaddon, K., Héloir, A., and Stricker, D. (2018). DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *3DV*.
- Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond. In *ICCV*.
- Malladi, R., Sethian, J., and Vemur, B. (1993). Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Mandikal1and, P. and Grauman, K. (2021). Learning dexterous grasping with object-centric visual affordances. In *ICRA*.
- Manhardt, F., Kehl, W., Navab, N., and Tombari, F. (2018). Deep model-based 6d pose refinement in RGB. In *ECCV*.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*.
- Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. In *Proceedings of the Royal Society of London*.
- Melax, S., Keselman, L., and Orsten, S. (2013). Dynamics based 3d skeletal hand tracking. *Graphics Interface*.
- Merriam-Webster dictionary (2021). Object.
- Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*.
- Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019). HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.
- Miller, A. and Allen, P. (2004). Graspit!: A versatile simulator for grasp analysis. In *Robotics Automation Magazine*.
- Moon, G., Chang, J., and Lee, K. M. (2018). V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*.

- Moon, G. and Lee, K. M. (2020). I2L-MeshNet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single RGB image. In *ECCV*.
- Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate non-iterative $O(n)$ solution to the pnp problem. In *ICCV*.
- Mottaghi, R., Xiang, Y., and Savarese, S. (2015). A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *CVPR*.
- Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2018). GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*.
- Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. (2017). Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *ICCV*.
- Mulcahy, N. J. and Call, J. (2006). Apes save tools for future use. *Science*.
- Mundy, J. L. (2006). Object recognition in the geometric era: A retrospective. In *Toward Category-Level Object Recognition*, Lecture Notes in Computer Science.
- Muron, W. (2020). motpy. <https://github.com/wmuron/motpy>.
- Nagarajan, T., Feichtenhofer, C., and Grauman, K. (2019). Grounded human-object interaction hotspots from video. In *ICCV*.
- Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. (2006). Laplacian mesh optimization. In *GRAPHITE*.
- Neverova, N., Thewlis, J., Guler, R. A., Kokkinos, I., and Vedaldi, A. (2019). Slim densepose: Thrifty learning from sparse annotations and motion cues. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nikolenko, S. I. (2019). Synthetic data for deep learning. *arXiv:1909.11512*.
- Niu, C., Li, J., and Xu, K. (2018). Im2struct: Recovering 3d shape structure from a single rgb image. In *Computer Vision and Pattern Recognition (CVPR)*.
- Nof, S. Y. (1999). *Handbook of Industrial Robotics*. John Wiley & Sons, Inc., 2nd edition.
- Oberweger, M., Rad, M., and Lepetit, V. (2018). Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *ECCV*.

- Oberweger, M., Wohlhart, P., and Lepetit, V. (2015a). Hands deep in deep learning for hand pose estimation. *Proc. Computer Vision Winter Workshop*.
- Oberweger, M., Wohlhart, P., and Lepetit, V. (2015b). Training a feedback loop for hand pose estimation. In *ICCV*.
- Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011a). Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*.
- Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011b). Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*.
- Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2012). Tracking the articulated motion of two strongly interacting hands. In *CVPR*.
- Olson, C. F. and Huttenlocher, D. P. (1997). Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*.
- Panteleris, P. and Argyros, A. (2017). Back to RGB: 3D tracking of hands and hand-object interactions based on short-baseline stereo. In *ICCV Workshops*.
- Panteleris, P., Kyriazis, N., and Argyros, A. A. (2015). 3d tracking of human hands in interaction with unknown objects. In *BMVC*.
- Panteleris, P., Oikonomidis, I., and Argyros, A. (2018). Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*.
- Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019a). DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*.
- Park, K., Patten, T., and Vincze, M. (2019b). Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*.
- Paschalidou, D., Ulusoy, A. O., and Geiger, A. (2019). Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *CVPR*.
- Paschalidou, D., van Gool, L., and Geiger, A. (2020). Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *CVPR*.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

- Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019a). Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*.
- Pavlakos, G., Kolotouros, N., and Daniilidis, K. (2019b). Texturepose: Supervising human mesh estimation with texture consistency. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Pavlakos, G., Zhu, L., Zhou, X., and Daniilidis, K. (2018). Learning to estimate 3D human pose and shape from a single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Pavlo, D., Feichtenhofer, C., Grangier, D., and Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*.
- Peng, S., Liu, Y., Huang, Q., Zhou, X., and Bao, H. (2019). Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*.
- Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. (2018). Sfv: Reinforcement learning of physical skills from videos. *ACM Transaction on Graphics*.
- Pentland, A. (1986). Parts: Structured descriptions of shape. In *AAAI*.
- Peris, M., Matsuto, A., Martull, S., Ohkawa, Y., and Fukui, K. (2012). Towards a simulation driven stereo vision system. In *ICPR*.
- Perkins, W. A. (1978). A model-based vision system for industrial parts. *IEEE Transaction on Computers*.
- Petrie, H. and Darzentas, J. (2011). Part-based robot grasp planning from human demonstration. In *ICRA*.
- Petrie, H. and Darzentas, J. (2017). Older people and robotic technologies in the home: Perspectives from recent research literature. In *PETRA*.
- Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *ICCV*.
- Pham, T.-H., Kyriazis, N., Argyros, A. A., and Kheddar, A. (2018). Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Pinkall, U. and Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*.

- Pitteri, G., Bugeau, A., Ilic, S., and Lepetit, V. (2020). 3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings. In *ACCV*.
- Pitteri, G., Ilic, S., and Lepetit, V. (2019). Cornet: Generic 3d corners for 6d pose estimation of new objects without retraining. In *ICCV Workshops*.
- Prasad, M., Fitzgibbon, A., Zisserman, A., and Gool, L. v. (2010). Finding nemo: Deformable object class modelling using curve matching. *CVPR*.
- PrimeSense (2013). <https://en.wikipedia.org/wiki/PrimeSense>.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.
- Rad, M. and Lepetit, V. (2017). BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. (2020). Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*.
- Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. (2020). Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *CVPR*.
- Rehg, J. M. and Kanade, T. (1994). Visual tracking of high DOF articulated structures: an application to human hand tracking. In *ECCV*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.
- Rhodin, H., Salzmann, M., and Fua, P. (2018a). Unsupervised geometry-aware representation learning for 3D human pose estimation. In *The European Conference on Computer Vision (ECCV)*.
- Rhodin, H., Spörri, J., Katircioglu, I., Constantin, V., Meyer, F., Müller, E., Salzmann, M., and Fua, P. (2018b). Learning monocular 3D human pose estimation from multi-view images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Riegler, G., Ulusoy, A. O., and Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Rijpkema, H. and Girard, M. (1991). Computer animation of knowledge-based human grasping. In *SIGGRAPH*.
- Roberts, L. G. (1963). *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology.
- Robinette, K. M., Blackwell, S., Daanen, H., Boehmer, M., Fleming, S., Brill, T., Hoeferlin, D., and Burnsides, D. (2002). Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Technical report, US Air Force Research Laboratory.
- Romero, J., Kjellström, H., and Kragic, D. (2009). Monocular real-time 3d articulated hand pose estimation. In *IROS*.
- Romero, J., Kjellström, H., and Kragic, D. (2010). Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *ICRA*.
- Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*.
- Rong, Y., Shiratori, T., and Joo, H. (2020). Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*.
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *ICCV*.
- Rusinkiewicz, S., Hall-Holt, O., and Levoy, M. (2002). Real-time 3D model acquisition. *ACM Transactions on Graphics*.
- Sahbani, A., El-Khoury, S., and Bidaud, P. (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*.
- Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*.
- Satkin, S., Lin, J., and Hebert, M. (2018). Data-driven scene understanding from 3d models. In *BMVC*.

- Shalavi, G. (2020). Youtube trends report: How adversity is shaping culture. <https://www.thinkwithgoogle.com/consumer-insights/consumer-trends/2020-youtube-trends/>.
- Shan, D., Geng, J., Shu, M., and Fouhey, D. (2020). Understanding human hands in contact at internet scale. In *CVPR*.
- Shen, B., Xia, F., Li, C., Martin-Martin, R., Fan, L., Wang, G., Buch, S., D’Arpino, C., Srivastava, S., Tchapmi, L. P., Vainio, K., Fei-Fei, L., and Savarese, S. (2020). igibson, a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint*.
- Shotton, J., Fitzgibbon, A., Blake, A., Kipman, A., Finocchio, M., Moore, B., and Sharp, T. (2011). Real-time human pose recognition in parts from a single depth image. In *CVPR*.
- Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*.
- Smith, B., Wu, C., Wen, H., Peluse, Patrick Sheikh, Y., Hodgins, J., and Shiratori, T. (2020). Constraining dense hand surface tracking with elasticity. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*.
- Smith, R. (2008). Open dynamics engine.
- Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *CVPR*.
- Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., and Kautz, J. (2020). Weakly supervised 3d hand pose estimation via biomechanical constraints. In *ECCV*.
- Spurr, A., Song, J., Park, S., and Hilliges, O. (2018). Cross-modal deep variational hand pose estimation. In *CVPR*.
- Sridhar, S., Mueller, F., Oulasvirta, A., and Theobalt, C. (2015). Fast and robust hand tracking using detection-guided optimization. In *CVPR*.
- Sridhar, S., Mueller, F., Zollhoefer, M., Casas, D., Oulasvirta, A., and Theobalt, C. (2016). Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*.
- Sridhar, S., Oulasvirta, A., and Theobalt, C. (2013). Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*.
- Sridhar, S., Rhodin, H., Seidel, H.-P., Oulasvirta, A., and Theobalt, C. (2014). Real-time hand tracking using a sum of anisotropic gaussians mode. In *3DV*.

- Stenger, B., Mendonça, P., and Cipolla, R. (2001). Model-based 3d tracking of an articulated hand. In *CVPR*.
- Sternberg, R. J. and Sternberg, K. (2017). *Cognitive Psychology*. Wadsworth, Cengage Learning.
- Su, H., Qi, C. R., Li, Y., and Guibas, L. J. (2015). Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*.
- Sun, X., Wei, Y., Shuang, L., Tang, X., and Sun, J. (2015). Cascaded hand pose regression. In *CVPR*.
- Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., and Freeman, W. T. (2018). Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*.
- Sundermeyer, M., Marton, Z.-C., Durner, M., Brucker, M., and Triebel, R. (2018). Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*.
- Sweetser, E. (1990). *From Etymology to Pragmatics: Metaphorical and Cultural Aspects of Semantic Structure*. Cambridge Studies in Linguistics. Cambridge University Press.
- Tagliasacchi, A., Schroder, Matthias Tkach, A., Bouaziz, S., Botsch, M., and Pauly, M. (2015). Robust articulated-icp for real-time hand tracking. *Eurographics Symposium on Geometry Processing*.
- Taheri, O., Ghorbani, N., Black, M. J., and Tzionas, D. (2020). GRAB: A dataset of whole-body human grasping of objects. In *ECCV*.
- Tan, D. J., Cashman, T., Taylor, J., Fitzgibbon, A., Tarlow, D., Khamis, S., Izadi, S., and Shotton, J. (2016). Fits like a glove: Rapid and reliable hand shape personalization. In *CVPR*.
- Tatarchenko, M., Richter, S. R., Ranftl, R., Li, Z., Koltun, V., and Brox, T. (2019). What do single-view 3d reconstruction networks learn?
- Taylor, J., Bordeaux, L., Cashman, T., Corish, B., Keskin, C., Soto, E., Sweeney, D., Valentin, J., Luff, B., Topalian, A., Wood, E., Khamis, S., Kohli, P., Sharp, T., Izadi, S., Banks, R., Fitzgibbon, A., and Shotton, J. (2016). Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In *SIGGRAPH*.

- Tekin, B., Bogó, F., and Pollefeys, M. (2019). H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*.
- Tekin, B., Sinha, S., and Fua, P. (2018). Real-time seamless single shot 6D object pose prediction. In *CVPR*.
- Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M.-P., Faure, F., Magnenat-Thalmann, N., and Strasser, W. (2004). Collision detection for deformable objects. In *Eurographics*.
- Tkach, Anastasia Pauly, M. and Tagliasacchi, A. (2016). Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*.
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*.
- Tsoli, A. and Argyros, A. (2018). Joint 3D tracking of a deformable object in interaction with a hand. In *ECCV*.
- Tu, Z. and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Tulsiani, S., Su, H., Guibas, L. J., Efros, A. A., and Malik, J. (2016). Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *CVPR*.
- Tung, H.-Y., Tung, H.-W., Yumer, E., and Fragkiadaki, K. (2017). Self-supervised learning of motion capture. In *NeurIPS*.
- Tzionas, D. (2017). *Capturing Hand-Object Interaction and Reconstruction of Manipulated Objects*. PhD thesis, University of Bonn.
- Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., and Gall, J. (2016). Capturing hands in action using discriminative salient points and physics simulation. *IJCV*.
- Tzionas, D. and Gall, J. (2015). 3D object reconstruction from hand-object interactions. In *ICCV*.
- Ungureanu, D., Bogó, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T. J., Tekin, B., Schönberger, J. L., Olszta, P., and Pollefeys, M. (2020). Hololens 2 research mode as a tool for computer vision research. *arXiv preprint arXiv:2008.11239*.

- Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., and Schmid, C. (2018). BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*.
- Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *CVPR*.
- Varzi, A. C. (1997). Boundaries, continuity, and contact. *Noûs*.
- Wan, C., Probst, T., Van Gool, L., and Yao, A. (2018). Dense 3d regression for hand pose estimation. In *CVPR*.
- Wang, F. and Hauser, K. (2019). In-hand object scanning via rgb-d video segmentation. In *ICRA*.
- Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018). Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*.
- Wang, Y., Min, J., Zhang, J., Liu, Y., Xu, F., Dai, Q., and Chai, J. (2013). Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics*.
- Weichao Qiu, Fangwei Zhong, Y. Z. S. Q. Z. X. T. S. K. Y. W. A. Y. (2017). Unrealcv: Virtual worlds for computer vision. *ACM Multimedia Open Source Software Competition*.
- Weise, T., Wismer, T., Leibe, B., and Van Gool, L. (2011). Online loop closure for real-time interactive 3D scanning. *Computer Vision and Image Understanding*.
- Wohllhart, P. and Lepetit, V. (2015). Learning descriptors for object recognition and 3d pose estimation. In *CVPR*.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes.
- Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A. X., Guibas, L. J., and Su, H. (2020). SAPIEN: A simulated part-based interactive environment. In *CVPR*.
- Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., and Savarese, S. (2016). Objectnet3d: A large scale database for 3d object recognition. In *ECCV*.
- Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*.

- Xiang, Y. and Savarese, S. (2012). Estimating the aspect layout of object categories. In *CVPR*.
- Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2018). PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*.
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*.
- Xiao, J., Russel, B. C., and Torralba, A. (2012). Localizing 3d cuboids in single-view images. In *NeurIPS*.
- Xiao, Y., Qiu, X., Langlois, P., Aubry, M., and Marlet, R. (2019). Pose from shape: Deep pose estimation for arbitrary 3D objects. In *BMVC*.
- Xu, Q., Wang, W., Ceylan, D., Mech, R., , and Neumann, U. (2019). Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*.
- Yang, Y., Feng, C., Shen, Y., and Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*.
- Ye, Y. and Liu, K. (2012). Synthesis of detailed hand manipulations using contact sampling. In *SIGGRAPH*.
- Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*.
- Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J. Y., Lee, K. M., Molchanov, P., Kautz, J., Honari, S., Ge, L., Yuan, J., Chen, X., Wang, G., Yang, F., Akiyama, K., Wu, Y., Wan, Q., Madadi, M., Escalera, S., Li, S., Lee, D., Oikonomidis, I., Argyros, A., and Kim, T.-K. (2018). Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zachiotis, G., Andrikopoulos, G., Gomez, R., Nakamura, K., and Nikolakopoulos, G. (2018). A survey on the application trends of home service robotics. In *ROBIO*.
- Zakharov, S., Shugurov, I., and Ilic, S. (2019). Dpod: 6d pose object detector and refiner. In *CVPR*.
- Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., and Yang, Q. (2016). 3D hand pose tracking and estimation using stereo matching. *arXiv:1610.07214*.

- Zhang, J. Y., Pepose, S., Joo, H., Ramanan, D., Malik, J., and Kanazawa, A. (2020). Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*.
- Zhou, T., Brown, M., Snively, N., and Lowe, D. (2017). Unsupervised learning of depth and ego-motion from video. In *CVPR*.
- Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2019). On the continuity of rotation representations in neural networks. In *CVPR*.
- Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.
- Zia, M. Z., Stark, M., and Schindler, K. (2014). Towards scene understanding with detailed 3d object representations. *IJCV*.
- Zimmermann, C. and Brox, T. (2017). Learning to estimate 3D hand pose from single RGB images. In *ICCV*.
- Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., and Brox, T. (2019). Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*.
- Çalli, B., Singh, A., and Walsman, A. (2015). The ycb object and model set: Towards common benchmarks for manipulation research. In *ICAR*.