Georg Poier

Learning without Labeling for 3D Hand Pose Estimation

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisors

Prof. Dr. Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology

Prof. Dr. Antonis A. Argyros
Computer Science Department
University of Crete

Graz, Austria, March 2020

To Viky

You cannot learn
what you think you already know

Adapted from Epictetus (c. 55 – 135)

Abstract

Data-driven approaches have lead to great successes for a broad range of tasks, but such approaches typically require large amounts of annotated data. In particular, big successes have been enabled by a large number of workers spending a huge effort to annotate enough data for training the respective systems. For this work the playground is set by the task of 3D hand pose estimation. A task, which is difficult due to the hands' high degree of freedom, fast motion, the frequently occurring (self-)occlusions and the 3D nature of the target. Due to the difficulties the best available solutions for this task rely on a combination of data-driven prediction models and prior knowledge about the task in terms of manually defined hand models. However, even the largest scale training datasets for the data-driven part are unable to sufficiently cover all the variations which might occur at test time, and thus, the whole system is hampered by the limitations of the training data. The insufficiency of the training data coverage is essentially due to the mentioned difficulties of the task, which also make the manual labeling a tedious effort. Additionally, new training data is often required for a new application, *e.g.*, due to a new camera setup. If it would be possible, however, to learn hand pose estimation without labeling effort it would be much more accessible and could thus be applied to a larger range of applications. Hence, the arising question is whether we can reduce this labeling effort or even find a way to solve the task without any manual labeling effort.

In this thesis we subsume three distinct methods, bringing us closer to eliminating the requirement for labeled real data. We do so by tackling the problem from different aspects. We first investigate the potential of exploiting prior knowledge in a specialized hybrid method. We devise a combination of a data-driven and a model-based part, where the model-

based part is able to cope with the insufficiency of training data for the data-driven part by incorporating the uncertainties of the data-driven part in the optimization of the pose parameters. In the subsequent chapter we reduce the label hunger of a data-driven approach already during training, by exploiting unlabeled data. Specifically, we show that by learning to predict a different view of the captured hand solely from a low dimensional latent representation extracted from the input view, an end-to-end trained model is enforced to make the latent representation very specific to the pose of the hand without requiring any pose labels. For the third main contribution of this thesis, we draw from another, completely different, source of information – namely synthetic data. To exploit synthetic data we mitigate the domain gap between real and synthetic data by learning to map from the features of real data to the features of synthetic data mainly from non-corresponding, *i.e.* unlabeled, data. In this way the synthetic data can be exploited to learn a very accurate mapping from the latent feature space to the target pose representation also for real data.

We show that each of the three main contributions of this thesis yields increased label efficiency. We find, that by exploiting prior knowledge with the hybrid approach we can achieve similar results with significantly less samples. The improvement is especially significant for small numbers of labeled samples. When exploiting unlabeled samples by learning to predict different views, we find that the improvement for a small number of labeled samples is even stronger, and in general about one to two orders of magnitude less labeled samples are sufficient to achieve similar results. Moreover, when exploiting synthetic together with unlabeled data we are able to improve the result when employing only about 0.1% of the labeled real data, *i.e.*, three orders of magnitude less than the strong baseline. Given the large effort for labeling and its importance for the success of a system, we believe that the methods introduced in this thesis present a step towards easier adoption of such technology making it accessible to a larger number of people and a larger range of applications.

Kurzfassung

Sogenannte datengetriebene Ansätze haben sich für ein breites Aufgabenspektrum bewährt – diese erfordern in der Regel jedoch große Mengen an annotierten Daten. Größere Erfolge für unterschiedliche Aufgaben wurden häufig erst durch enormen menschlichen Annotierungsaufwand ermöglicht. Die Aufgabe an der wir die Ergebnisse dieser Dissertation messen ist die Schätzung der Hand-Pose in 3D. Dabei handelt es sich um eine Aufgabe deren Schwierigkeiten in der hohen Anzahl an Freiheitsgraden, schnellen Bewegungen, häufig auftretenden (Selbst-)Verdeckungen und der gewünschten 3D-Schätzung, liegen. Aufgrund dieser Schwierigkeiten basieren die besten Lösungen für diese Aufgabe auf einer Kombination datengetriebener Vorhersagemodelle und Vorwissen über die Handanatomie. Allerdings kann selbst der größte existierende Trainingsdatensatz für das datengetriebene Modell nicht alle Variationen abdecken, die zur Testzeit auftreten können. Die Unzulänglichkeiten der Trainingsdaten lassen sich im Wesentlichen auf die angesprochenen Schwierigkeiten der Aufgabe zurückführen, welche die manuelle Annotierung zu einem enormen Aufwand machen. Das ist umso problematischer da es für eine neue Anwendung häufig nötig ist auch neue Trainingsdaten zu generieren – z.B. weil für die neue Anwendung ein anderes Kamera-Setup erforderlich ist. Wenn es aber möglich wäre, ein System zur Schätzung der Hand-Pose ohne Annotierungsaufwand zu trainieren, wäre ein solches System viel leichter anwendbar, einem größeren – im speziellen weniger finanzkräftigen – Kreis an Entwicklern zugänglich und damit auch für viele neue Anwendungen verwendbar. Daher gehen wir in dieser Arbeit der Frage nach, ob es möglich ist, den Annotierungsaufwand zu reduzieren oder sogar einen Weg zu finden um die Aufgabe gänzlich ohne Annotierungsaufwand zu lösen.

Diese Arbeit fasst drei Methoden zusammen, die uns dem Ziel, für die Schätzung der Hand-Pose keine real-world Daten manuell annotieren zu müssen, näher bringen. In einem ersten Schritt untersuchen wir das Potenzial der Nutzung von Vorwissen über die Handanatomie mittels eines Handmodells. Dazu stellen wir eine Kombination aus einem daten- und einem modellbasierten Teil vor, in welcher die Schätz-Ungenauigkeiten des datengetriebenen Modells ausgeglichen werden, indem die Verteilung der Schätzwerte in der nachfolgenden Optimierung der Parameter des Handmodells explizit berücksichtigt wird. Im darauffolgenden Kapitel reduzieren wir den Annotierungsbedarf eines datengetriebenen Ansatzes bereits während des Trainings, indem wir auf nicht annotierte Daten zurückgreifen. Dazu trainieren wir ein Model um vorherzusagen wie die Hand aus einem anderen Blickwinkel aussieht. Wir zeigen dass unser – auf diese Weise – trainiertes Model eine latente Repräsentation beinhaltet, die sehr spezifisch für die Hand-Pose ist, ohne jedoch mit solchen Annotierungen trainiert zu werden. Für den dritten Hauptbeitrag dieser Arbeit ziehen wir wiederum eine völlig andere Datenquelle heran – nämlich synthetische Daten. Damit synthetische Daten trotz deren Unterschiede zu realen Daten besser ausgenutzt werden können, lernen wir eine Abbildung von der latenten Repräsentation realer Daten auf die latente Repräsentation synthetischer Daten vorwiegend aus nicht korrespondierenden, d.h. nicht annotierten Daten. Auf diese Weise können die synthetischen Daten, mittels derer ein äußerst akkurates Schätzmodel gelernt werden kann, auch für die Schätzung der Posen in realen Daten ausgenutzt werden.

Es zeigt sich, dass jeder der drei Hauptbeiträge dieser Arbeit den Annotierungsaufwand verringert. Durch die Ausnutzung von Vorwissen mittels des hybriden Ansatzes zeigt sich eine deutliche Verbesserung speziell bei einer kleinen Anzahl zur Verfügung stehender annotierter Beispiele. Durch die Nutzung nicht annotierter Daten von anderen Blickwinkeln während des Trainings lassen sich ähnliche Ergebnisse bereits mit 10 bis 100-mal weniger annotierten Trainingsbeispielen erzielen. Durch die Verwendung von synthetischen zusammen mit nicht annotierten Daten verbessert sich das Ergebnis schließlich schon mit etwa 0,1% der annotierten realen Daten. Angesichts des großen Annotierungsaufwands und dessen kritischer Bedeutung für den Erfolg eines solchen datengetriebenen Systems sehen wir die in dieser Arbeit vorgestellten Methoden als einen Schritt in Richtung der einfacheren Anwendbarkeit einer solchen Technologie, womit sie gleichzeitig einer größeren Anzahl von Menschen und weiteren Anwendungsbereichen zugänglich wird.

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.

The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date

Signature

Acknowledgments

Disclaimer: This section clearly fails to fulfill its purpose.

Thinking about the people, which deserve acknowledgment, it became clear that the number of people without whom this thesis would not exist in the form it does or not exist at all is incredibly large. This starts with the people which inspired me and raised my interest in various topics underlying this thesis, shaped my way of thinking, or taught me the various basics, which turned out to be essential to complete the thesis. But it is also about the people upon whose work I could build, the people which supported me in various ways, or those who made me keep up the motivation over the years. Moreover, this includes the many people incorporated in providing financial support, enabling me to do this while being employed at the institute, which overall, enabled a reasonable balance between project work and research, not requiring me to hunt for totally unrelated financial resources.

Naming all the people upon whose work I built by using their tools, their code or by drawing inspiration from their ideas, altogether crucially shaping this thesis, would alone become a virtually impossible endeavor. But, even when omitting more "distant contributors" a proper acknowledgment would easily go beyond the reasonable constraints or otherwise make me feel that nobody would actually be properly acknowledged. That is,

this is still a failure.

Given the above disclaimer, I try my best to name some of the most crucial contributors and apologize in advance to the many I miss. I thank Horst Bischof for giving me the opportunity to pursue a PhD while working at the institute and at the same time giving me the freedom to explore my own ideas. Equally important I am grateful to Horst for making the institute the fruitful and pleasant working environment it is. I thank Antonis Argyros for giving me the chance to intern in his research group before starting the PhD endeavor. The internship largely raised my desire for doing further research and hence, feel certain that doing a PhD is the right path for me. Not least because of that I am also grateful to Antonis for agreeing to act as the second supervisor for this thesis.

Moreover, I want to specifically thank the members of our closer research group – meanwhile well led by Horst Possegger – including a rather large number of current and former office mates. The collaboration and support in many ways, as well as funny social activities contributed strongly to making the time of my PhD studies an enjoyable period in my life. The same is valid for many current and former members of the other groups at the institute – especially for countless inspirational discussions as well as funny distractions from work, *e.g.*, at reading groups, beers for X, (coffee-) breaks, five o'clock teas, skiing-trips, soccer- and, importantly, table soccer games.

Furthermore, I am grateful to my current and former colleagues at FORTH, the 3D-Pitoti Project, IGT, ivii and Microsoft Research for their collaboration, support and inspiration. These shaped this thesis in many different ways.

Finally I would like to thank my family and friends outside the working environment – including those that are not able to read this anymore but will never be forgotten. I am especially grateful for the support throughout the years, in stressful and less stressful times. This made me keep up the motivation to ultimately finish.

Danke!

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 What is the problem?

Successful approaches to learning models for hand pose estimation require extensive amounts of labeled data. This is not only the case for hand pose estimation, but equally holds for many other classic computer vision tasks, such as object detection, semantic segmentation, or image classification. All of these tasks have seen a significant performance increase in recent years following the availability of large annotated datasets and the corresponding computational power to exploit these large labeled datasets.

In general, the task of providing labeled data is expensive and has thus even become an industry of its own[1]. This especially holds for tasks for which the target space is structured in a complex manner, like for semantic segmentation, human pose estimation or the task considered in this work: hand pose estimation.

[1] The probably most prominent example for the labeling industry is Amazon Mechanical Turk (`www.mturk.com`). Nevertheless, many specialized companies exist: *e.g.*, cloudfactory (`www.cloudfactory.com`), Might Ai (`www.mighty.ai`) or DataPure (`www.datapure.co`).

Considering 3D hand pose estimation, the labeling is effortful due to the frequent self occlusions, the 3D nature of the hand and targets, and especially the large pose space. The pose space grows exponentially with the degrees of freedom (DoFs) of the hand pose. Model based works on hand pose estimation usually parameterize the pose of their hand models with at least 26 DoFs (Oikonomidis et al., 2011a; Taylor et al., 2016; Xu and Cheng, 2013). Hence, when trying to illustrate the size of the emerging space by taking just three samples from each dimension and ignoring 3D translations, *i.e.*, using 23 DoFs instead of 26 DoFs, we end up with $3^{23} \approx 9.4 \times 10^{10}$, *i.e.*, 94 billion poses. While such a rough calculation has flaws[2], it should merely serve as an indication of the size of the target space and why even the largest datasets to date are far from capturing the space well.

Moreover, each time specific assumptions underlying the task change, it is likely that a large amount of new labeled data has to be provided. This might be the case if the nature of occlusions changes, *e.g.*, if the hand pose should also be estimated during interaction with objects or different objects are interacted with. Other examples for cases where new data is commonly required are, *e.g.*, when a new view point is targeted, (*c.f.*, data from an ego-centric view vs. third-person view) or a new sensor with different noise characteristics is to be used. Whatever the reasons for such changes might be, the necessary labeling effort hampers applications.

In this thesis we develop methods which largely reduce the labeling effort. Before introducing these methods we want to point out why the specific contributions are necessary. Hence, we will first briefly discuss the main approaches towards hand pose estimation and will especially point out the limitations which make the contributions of our work necessary.

1.2 Hand pose estimation: a brief review

Traditionally, works on hand pose estimation are categorized into two main strands (Erol et al., 2007; Supancic et al., 2015; Wu et al., 2001). These two main strands are often denoted *model-based* and *data-driven*. Model-based approaches fit a manually created hand model to each observation. Data-driven approaches, on the other hand, learn a mapping from input to target pose from data and apply this mapping to each observation. Finally, *hybrids*, which combine ideas from both strands, have been developed.

[2] The calculation ignores, *e.g.*, anatomical and physical constraints reducing the space, but also that several of the DoFs will not be sufficiently captured with only three samples

Model-based paradigm

Early works on hand pose estimation usually followed the model-based paradigm (Kuch and Huang, 1995; Rehg and Kanade, 1995; Wu and Huang, 1999). The main observation is that prior knowledge about the task and especially the hand can be exploited to find the pose which generated the image. With respect to our work, exploiting prior knowledge also circumvents the requirement for large datasets.

To estimate the pose of the hand, approaches from this strand follow an *analysis-by-synthesis* approach (Colman, 2015). In these approaches a manually crafted hand model, which can be compared to the observation, is employed. During test time, the task is to find a parameterization of the model, so that the model best fits the observation. The fit can be judged by, *e.g.*, rendering the parameterized model and comparing it with the actual observation. By exploiting prior knowledge in this way, such approaches do not only eliminate the requirement of a large annotated dataset, the estimated poses can also be guaranteed to always be physically plausible.

Since computational constraints render it infeasible to exhaustively search the pose space for the best parameterization, such approaches require an initialization, which is already close to the true solution. Hence, the approaches typically rely on an initialization based on the solution from the previous frame and on a manual initialization in the first frame. However, especially fast hand movements or dropped frames can lead to tracking errors, from which such a tracking based approach can usually not recover.

Data-driven paradigm

The interest in data-driven methods for hand pose estimation raised within the last five to ten years (*c.f.*, (Erol et al., 2007; Yuan et al., 2018)). This goes together with a generally increased interest in data-driven methods and their increased performance, which was largely triggered by increased computational resources, together with the construction and availability of larger datasets.

Data-driven methods learn a mapping from the input to the pose based on annotated training data. Such a mapping is a complex, highly non-linear function. Hence, learning the mapping from data – for a large range of poses – requires a large amount of labeled data. The labeled data is then employed to learn the mapping function during usually lengthy training times. During test time the learned functions can then quickly deliver a pose estimate.

Providing a sufficient amount of training data can, however, be a large effort as described above. Additionally, these approaches build on the assumption that the poses encountered during test time are covered by the pose distribution in the training set, since the approaches are prone to fail for poses outside the training set distribution. However, – as described above – datasets of current scale are still far from capturing all possible poses.

Hybrid approaches

We see that model-based and data-driven approaches have their individual merits and drawbacks. Hence, hybrid approaches aiming to combine the merits and omit the drawbacks of both strands have been developed.

Hybrid approaches usually produce an initialization using a data-driven element and validate and/or locally optimize the initial solution using a model-based element (Panteleris et al., 2018; Sharp et al., 2015; Ye et al., 2016). Nevertheless, some hybrid approaches mainly rely on the model-based element (Ganapathi et al., 2010; Wei et al., 2012) and use the data-driven part only for error correction.

While some merits of both strands can be exploited and some drawbacks overcome, important issues are not fully circumvented. These approaches are typically able to always provide a physically valid solution, while not requiring a manual initialization in the first frame, nor requiring the solution in the previous frame to be correct or close to the solution in the current frame. Additionally – as we show in Chapter 3 – the sampling density of the labeled training set can be reduced. Nevertheless, hybrid approaches usually still rely on a training set, which densely covers the pose distribution encountered at test time. This is due to the fact that – at least – for problem cases of the model-based part, like initialization or track losses, the estimate of the data-driven part needs to be close to the true solution. The data-driven approach, however, will not be able to deliver an initial solution close to the true solution if the training set does not reasonably well cover the test distribution. Hence, hybrid approaches still depend on the coverage of the training set.

1.3 Towards closing the gaps

The quest of this thesis is to reduce the labeling effort for learning models for hand pose estimation while not sacrificing accuracy. To investigate this question, we essentially explore three directions: (i) exploiting prior knowledge about the hand, (ii) exploiting unlabeled data and (iii) exploiting synthetic data. The methodological contributions towards these three directions, which we introduce in this thesis, are largely based on three previous conference publications (Poier et al., 2019, 2015, 2018) and contain additional experiments and analyses.

Naturally, a model learned from a sparsely sampled training set would be prone to errors. One way to mitigate this is to exploit prior knowledge about the structure of the hand. Such prior knowledge can, *e.g.*, be provided by employing a graphical model of the hand.

Combining a learned/data-driven and a hand-model-based part in a hybrid approach reduces the need for a densely sampled training set. In Chapter 3 we devise such a hybrid approach, where we first learn a model, which is able to make independent estimates for each joint location and in this way also lessens the requirement to densely cover the pose space. During test time, after applying the learned model, we employ a graphical hand-model to fit the independent joint estimates of the learned model.

By fitting the hand-model, the system always delivers an anatomically correct hand pose, even if the independent joint position estimates from the learned model would not form an anatomically plausible pose. Moreover, by having the learned model outputting a distribution of joint positions and optimizing the hand-model parameters by considering this distribution, the hand-model-based step considers the uncertainties of the learned model when optimizing the hand-model parameters to provide the final result. We show that in this way we can achieve similar accuracy using significantly less labeled data.

Despite the discussed advantages of our hybrid approach, it still requires a significant amount of labeled training data. This is because the model-based step cannot fix gross errors that occur for samples to far from the training set distribution.

In quest of having the training data better covering the pose space without requiring more labeling effort, we develop a way to exploit unlabeled data. In Chapter 4 we show that unlabeled data can provide supervision for the task of pose estimation if two cameras observe the hand simultaneously from different viewpoints. The idea is based on the observation that the pose is predictive for the appearance of the hand from any known view. That is, given the pose, the hands' appearance can be roughly predicted from any known view. Based on this observation we show that representations, very specific to the hand pose, can be learned by simply learning to predict one camera view, given the other.

More specifically, given the image from one camera view, we first learn to predict a low dimensional latent representation. Intuitively, if this latent representation contains detailed information about the pose of the hand, a second model should be able to predict another view of the hand – given only the latent representation. By training the two models jointly we show that the latent representation becomes pose specific and – when trained with labeled data – the system needs one order of magnitude less labeled samples to learn to map from the latent representation to the target pose with similar or better accuracy.

Exploiting unlabeled data can reduce the labeling effort, but, learned models still need labeled data to define the target space. The latent representation learned only from unlabeled data is by definition agnostic to the exact target representation we are aiming at, like 3D joint positions or joint angles. Hence, we always need labeled samples to learn the final mapping to the desired target space.

For hand pose estimation – similar to other tasks – we can also render a virtual model to generate labeled synthetic data, which can then potentially be exploited to define the target space. This has the advantage that a virtually infinite amount of training data together with accurate annotations can be generated easily. Nevertheless, when aiming to exploit synthetic data we have to overcome the domain gap between synthetic and real data. While, this can be overcome using corresponding real and synthetic data and learning to map from the one to the other,the correspondence between real and synthetic data needs to be established in the pose space, and hence, pose labeled real data is required for such an approach.

In Chapter 5 we develop a way to overcome the domain gap between real and synthetic data by learning a joint feature space of real and synthetic data using mainly unlabeled data. The idea is to ensure that the feature distributions of real and synthetic data are aligned using an adversarial training loss term and simultaneously ensure that similar poses are mapped to similar locations in the latent feature space by enforcing pose specificity in the feature space using the view prediction loss described above. Using these contributions the system improves the results of the strong baseline with only about 0.1% of the labeled real data and achieves results in the range of the state-of-the-art approaches when training with only about 1% of the labeled real data they use.

CHAPTER 2

Towards learning without manual supervision: the background

Contents

The intention of this chapter is to provide the relevant background for our contributions. To have the thesis self contained we start by defining what we mean when we talk about a system which learns. We subsequently transition our discussion towards concepts which are increasingly specific to our task. This includes a discussion of important strands of work exploiting unlabeled data. The first focus of this chapter will be on *self-supervised learning* to exploit unlabeled data (Section 2.3). We discuss self-supervised learning strategies in more detail as they are relevant for the contributions in two of the three methodological chapters of this thesis. In the sequel we discuss research on transfer learning (2.4), an area which is relevant for crucial parts of the contributions in Chapter 5, where we aim to overcome the domain gap between synthetic and real data. The final part (2.5) of this chapter deals with approaches exploiting prior knowledge in terms of hand models and is thus most specific to our task of hand pose estimation. A more detailed discussion of approaches which are closely related to our contributions – especially in the context of the underlying publications – are provided in the chapters presenting the respective contributions.

2.1 Notation

To facilitate reading this thesis we briefly summarize the notation and definitions we use throughout the work. For convenience Table 2.1 provides a quick reference of the notations.

As in many modern scientific works in our field, we use lowercase italicized Latin or Greek letters for scalars, *e.g.*, i, α, ω. If not stated otherwise, we employ uppercase italicized Latin letters to specifically denote scalar constants, like D, J, N. Bold letters denote vectors or matrices, *e.g.*, $\mathbf{x} = (x_1, \ldots, x_N)^\top$. Vectors are assumed to be elements of a real valued vector space $\mathbb{R}^N$, for which we use an uppercase blackboard bold letter with the exponent describing the dimensionality of the space. To keep the notation general, similar to vectors, also images are usually described by lowercase bold letters. In this case the dimensionality of the vector space is simply given by the product of the height, H, width, W, and number of channels, C, of the image, *i.e.*, $\mathbf{y} \in \mathbb{R}^D$, where $D = H \times W \times C$. Furthermore, we use uppercase calligraphic letters to denote sets, like a set of images, $\mathcal{I} = \{\mathbf{x}_i\}_{i \in \{1,\ldots,N\}}$, of a specific image size: $\mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$. The size of the set $\mathcal{I}$, *i.e.*, its cardinality, is given by $N = |\mathcal{I}|$. Functions, mapping between different vector spaces, are represented by lowercase italicized letters, *e.g.*, to describe the result of a function f mapping from an N- to a D-dimensional space, *i.e.*, $f\colon \mathbb{R}^N \to \mathbb{R}^D$ we write $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^D$. As in the previous example, for the sake of notational clarity we usually omit the parameters on which such functions depend. Nevertheless, sometimes it makes sense to make such parameters explicit. In this case, if the output of a function f depends on parameters $\boldsymbol{\theta}$, we write $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$. Sometimes we aim to achieve a clearer notation by concatenating corresponding, individual elements within a single entity, denoted tuple. We represent a tuple by a Greek uppercase letter, *e.g.*, $\Upsilon = (\mathbf{x}, c)$. Finally, we also follow common practice by having $\|\cdot\|_p$ represent the L^p-norm of the argument, *i.e.*, $\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$.

For the sake of readability we refer to *all* 3D positions, which are used to define the pose of the hand, by *joint locations*. That is, for the sake of readability, this can include positions, which are not actually joints, like the center of the palm or finger tip positions.

Table 2.1: **List of notations.** Notational conventions for commonly used entities.

Entity	Example notation
Scalar	i, α, ω
Constant	D, J, N
Vector	$\mathbf{x} = (x_1, \ldots, x_N)^\top$
N-dimensional vector space	$\mathbb{R}^N$
Image	$\mathbf{y} \in \mathbb{R}^{W \times H \times C}$
Set	$\mathcal{S}$
Function (generic)	$f \colon \mathbb{R}^N \to \mathbb{R}^D$
Function (notation with explicit parameters $\boldsymbol{\theta}$)	$f(\cdot; \boldsymbol{\theta})$
Tuple	$\Upsilon = (\mathbf{x}, c)$

2.2 Learning from data

The theory for learning machines goes back to at least the mid of the 20th century[1] and was at that time strongly connected with the question whether machines can think. The question about the possibility to build intelligent machines was discussed prominently (Jefferson, 1949; Turing, 1950) and first experiments were conducted (*c.f.*, McCarthy and Feigenbaum (1990); Prinz (1988); Strachey (1952)). While the quest of our work is a rather different one, the way to approach it is very related: by creating a system to learn autonomously.

In a seminal work Alan Turing proposed to replace the prominent question whether machines can think by rather asking whether a machine could well imitate a human (Turing, 1950). In this work, Turing proposed the "imitation game" to test whether a machine is able to imitate a human. The proposed test became well known as the *Turing test*. In the same work Turing also discussed how a machine could be implemented to succeed in his test. One of his conjectures was that a machine – in order to call it "intelligent" – has to depart from the regime where it does exactly what it is ordered to do, but rather has to learn autonomously to some extent. Such an autonomous learning could be based on, *e.g.*, some supervision signals from a teacher, but also from other sources without explicit supervision.

In line with Turing's approach to define an "intelligent" machine by focusing on what it can do, Tom Mitchell later formulated the probably most cited definition of a machine learning algorithm (Mitchell, 1997): "A computer program is said to learn from experience E with respect to some

[1] Some might note that theoretical considerations go back to at least ancient Greek and Chinese tales (McCorduck, 2004). Unfortunately, a discussion of the relations to these considerations is outside the scope of our work.

class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." In our case the task would be hand pose estimation for which a performance measure could be the distance between estimated and ground truth positions on some test set. According to Mitchell's definition such a hand pose estimation system is said to learn if the distance between its estimates and the corresponding ground truth on the test set becomes smaller after having been provided more training data.

For current machine learning based hand pose estimation systems, the experience in Mitchell's definition is usually provided by labeled training data. Such labeled data for hand pose estimation are tuples $\Upsilon = (\mathbf{x}, \mathbf{y})$ consisting of data samples, $\mathbf{x}$, and their corresponding labels, $\mathbf{y}$. The data samples are images showing a hand, and the labels define the pose of the hand.

However, the goal of this work is to learn a model for hand pose estimation with as little manual supervision as possible, *i.e.*, the experience in Mitchell's definition should require very little manual input. This means that pose labels are, *e.g.*, provided only for a small number of real data samples, since real samples can only be accurately labeled with significant manual effort.

The system should instead exploit other sources of information, which do not require manual input on a per frame basis. For example, exploiting some form of contextual cues can help to learn how visual entities from different frames relate to each other without needing to rely on manual supervision. This can be related to the conjectures of Turing (1950) mentioned above and (if desired) also to learning in biological systems, which are assumed to learn visual recognition in a rather self-supervised way in their infancy (Held and Hein, 1963; Markman, 1991; Nawrot et al., 2009) and rely more and more on (learned) models later (Decker et al., 2015; Hartley and Somerville, 2015; Kahneman, 2011).

For data for which we do not have labels available we employ supervision signals which are derived from different parts of the input. This is related to a strand of works for which the corresponding learning tasks are usually formulated so that the system has to learn to relate visual entities to each other to solve the task. Such relations can be learned based on the recognition of typical positions and/or orientations of object parts within an image (Doersch et al., 2015; Gidaris et al., 2018; Noroozi and Favaro, 2016), across frames (Sümer et al., 2017; Wang and Gupta, 2015; Wang et al., 2019b), or between different modalities (Dosovitskiy and Koltun, 2017; Owens et al., 2016) and domains (Huang et al., 2018; Massa et al., 2016; Pratt, 1992).

This form of weak supervision is often called *self-supervision* or *natural supervision*. We follow this naming in this work.

That is, the learning tasks considered in this work are always based on some form of supervision – even if no target labels are available. Hence, as usual for such supervised learning tasks, learning accounts for adjusting the parameters $\boldsymbol{\theta}$ of a function f in a way, that the output $\hat{\mathbf{y}}$ of the function for a given sample $\mathbf{x}$,

$$\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta}) \tag{2.1}$$

corresponds to the samples' label $\mathbf{y}$. The label $\mathbf{y}$, which is used as supervision signal during training, can be the desired target label if it is available, but can also represent, *e.g.*, a "surrogate label" as it is the case for typical self-supervised approaches. We will show examples for such surrogate labels in Section 2.3.

For such a supervised learning task the function parameters are found by minimizing a loss ℓ over a training dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{|\mathcal{S}|} \ell\left(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i\right) \tag{2.2}$$

That is, by using surrogate labels even the exploitation of unlabeled data can be formulated in the same way as standard supervised learning tasks. Nevertheless, many approaches have been proposed, which exploit unlabeled data without relying on supervision signals in this sense (Hastie et al., 2009). Most notably in this respect are unsupervised and semi-supervised learning approaches. In the following we thus briefly discuss these prominent strands and put our work in perspective with regard to them.

2.2.1 Unsupervised learning

An important ingredient of this thesis is the exploitation of unlabeled data. Learning from unlabeled data is often used interchangeably with *unsupervised learning* (Doersch et al., 2015; Dosovitskiy et al., 2014; Garg et al., 2016). However, the usage of the term unsupervised learning is not always very clear and consistent throughout different works (Chapelle et al., 2006; Dosovitskiy et al., 2014; Goyal et al., 2019). We, thus, briefly discuss the term here.

Unsupervised learning obviously relies on exploiting unlabeled data. That is, the training dataset $\mathcal{X}$ contains only the data samples $\mathbf{x} \in \mathcal{X}$, without any corresponding supervision information like labels. The goal of unsupervised learning is usually described as finding interesting structure in the data. This search for structure is often related to the problem of *density* *estimation* (Hastie et al., 2009; Hinton et al., 1999). Likewise, prominent approaches to unsupervised learning, like clustering (Comaniciu and Meer, 2002; Lloyd, 1982), or dimensionality reduction (Hotelling, 1933), estimate a known functional of the density (Chapelle et al., 2006). Consequently, some researchers argue that *density estimation* would be a better name for unsupervised learning (Chapelle et al., 2006).

To circumvent misconceptions arising from the aforementioned inconsistency in the usage of the term unsupervised learning and the way we exploit unlabeled data in this work, we will avoid using the term *unsupervised learning* for our work. In this work we exploit unlabeled data to learn image representations. For the learning task, however, we do not have any target supervision. Instead we exploit supervision signals, which can be naturally derived from the data itself. We discuss specifically related approaches, *i.e.*, approaches which learn representations usually without having supervision for these representations, in the subsequent sections (esp. in Sections 2.3 and 2.4).

2.2.2 Semi-supervised learning

Most often, in this thesis we exploit unlabeled data together with labeled data. Approaches, which learn models by exploiting labeled and unlabeled data together, are usually coined *semi-supervised*.

The various existing semi-supervised learning approaches are inspired by several different but related assumptions. Examples for such assumptions are that data points are likely to belong to the same class if they belong to the same cluster (Ji et al., 2018; Zhu et al., 2003), that the decision boundary should be in a region of low density (Joachims, 1999; Vapnik, 1998), or that the high dimensional data lies on a low dimensional manifold, which can be viewed as an approximation of the high-density regions (Chapelle et al., 2006).

The most common assumptions for semi-supervised learning can be subsumed by the so-called *semi-supervised smoothness assumption* (Chapelle et al., 2006). It states that for two data points $\mathbf{x}_1$ and $\mathbf{x}_2$, which are close in a high-density region, the corresponding predictions $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$ should also be close. Figure 2.1 shows a prominent toy dataset to illustrate why such an assumption can yield a decision boundary with intuitively better generalization capabilities when exploiting unlabeled data. While, in general, the assumption applies to classification *and* regression tasks, common instantiations of the *cluster assumption*, or the *low-density separation* mentioned above are less useful for regression tasks (*c.f.*, Grandvalet and Bengio (2004); Laine and Aila (2017); Lee (2013)).

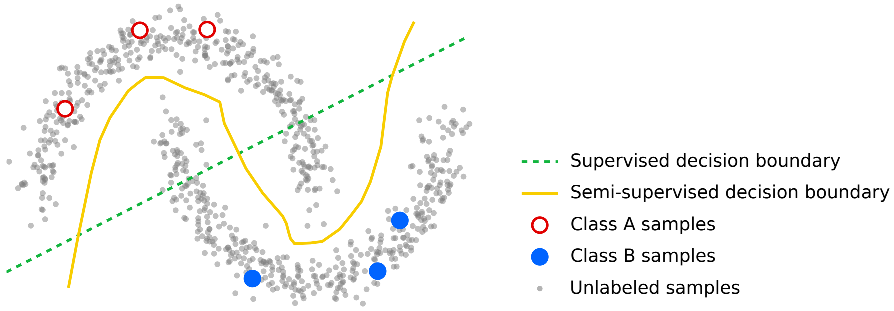


Figure 2.1: **Intuition for semi-supervised learning.** Illustration of decision boundaries on the prominent *two moons* dataset. The dashed green line illustrates a decision boundary of a supervised approach. The solid yellow line is the decision boundary which can be obtained by an adequate semi-supervised model considering the unlabeled data – like the Π-*Model* (Laine and Aila, 2017), or the *Mean Teacher* (Tarvainen and Valpola, 2017). See *e.g.*, Oliver et al. (2018) for a comparison of specific decision boundaries of different approaches on this dataset.

In line with this, the regression methods presented in Chapter 4 and 5 are most related to approaches relying on the manifold assumption. Following Chapelle et al. (2006), a well trained manifold should prove more useful for regression tasks. This becomes more intuitive when assuming the manifold to be an approximation of the high density regions. In this case the predictions $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$ should be close for *any* two data points $\mathbf{x}_1$ and $\mathbf{x}_2$ which are close on the manifold. This essentially describes the *smoothness assumption of supervised learning*. That is, since the manifold is already an approximation of the high density regions, on the manifold, the "closeness constraint“ is not restricted to high-density regions anymore and the semi-supervised smoothness assumption reduces to the smoothness assumption of supervised learning. This means that we can apply standard supervised learning on the manifold, if we can assume a proper manifold.

To ensure a well trained manifold in Chapter 4 and 5 we add additional loss terms similar to many prominent state-of-the-art approaches (Berthelot et al., 2019; Oliver et al., 2018; Tarvainen and Valpola, 2017). In our case these loss terms essentially implement a self-supervised approach as well as an approach to domain adaptation. Therefore, we will focus on a review of the related work for these two strands in the subsequent sections.

2.3 Self-supervised learning

The term self-supervision is often used to denote a way of supervision, where the supervision signals can be generated automatically from the observations (de Sa, 1993; Dosovitskiy and Koltun, 2017; Schmidhuber and

Prelinger, 1993). Such supervision signals can, *e.g.*, be "natural" text sequences in a document, sound corresponding to an image, or the subsequent frames in a video. Since the input and supervision signals are "naturally" paired, sometimes the term *natural supervision* is used for such approaches (Dosovitskiy and Koltun, 2017; Gomez-Bigorda et al., 2017). In this section we discuss the background on this topic in more detail, since such ideas are crucial for two of the three methodical chapters of this thesis.

Early works exploit paired sensory inputs (de Sa, 1993; Yamauchi et al., 1999) or the context of text (Schmidhuber and Prelinger, 1993). For instance, they use image or video data with the corresponding sound (de Sa, 1993; Yamauchi et al., 1999) and train a network for each modality by minimizing the disagreement between the output of both networks for a paired sample. Schmidhuber and Prelinger (1993) proposed a similar loss for two networks, *e.g.*, operating on disjoint parts of the same textual sentence. In this case, one network is supposed to predict a possibly abstract class label for its text input, the other network should predict the same class label as the first network, but is only given the context of the input to the first network.

Similar ideas are seized by many prominent recent works. In the text domain, learning to predict a word from its context has been shown to yield powerful representations (Mikolov et al., 2013a,b). These ideas were later also transfered to the image domain. Instances of the proposed tasks consisted of learning to predict where the relative location of image parts (Doersch et al., 2015) or inpainting large parts of an image just based on the remaining context (Pathak et al., 2016).

In general such tasks are designed so that the model has to learn to relate parts of the input, *i.e.*, which parts belong to an object and/or in which relation the parts are. Since such tasks are usually different from the final goal (*e.g.*, text understanding or object recognition), they are often coined *surrogate*, *pretext* or *proxy tasks*. A crucial advantage of such tasks is that the input and the corresponding supervision signal are "naturally" paired within, *e.g.*, a text document, image or video. That is, no expensive labeling is necessary.

In the following we discuss such self-supervised tasks from different areas of research in more detail. To provide appropriate context, we start our discussion with prominent examples from the language modeling and computer vision literature, and subsequently focus on approaches which are specifically related to our work in terms of the input cues they exploit (*i.e.*, multi-view) and the task they target (*i.e.*, pose estimation).

2.3.1 Learning language models

For learning language models, self-supervised approaches have been very successful (Devlin et al., 2018; Mikolov et al., 2013b). Predicting a masked word, sentence or paragraph given its context from a structured document (or vice versa (Mikolov et al., 2013b)) became a standard task for learning language models. Some very prominent approaches from this domain are *word2vec* (Mikolov et al., 2013b), *ELMo* (Peters et al., 2018), *GPT* (Radford et al., 2018) and *BERT* (Devlin et al., 2018).

In word2vec (Mikolov et al., 2013b) the surrogate task is – given a word – to predict the surrounding words within a document. For this task, learning is posed as a classification problem. That is, each possible target word is given a corresponding class label and the goal is to predict the class labels of the words to be predicted. In this case the number of classes is essentially equal to the number of words in a vocabulary and thus often very large. For more efficient training a variant of Noise Contrastive Estimation (NCE) (Dyer, 2014; Gutmann and Hyvärinen, 2012) is used. For NCE the correct word has to be discriminated from randomly sampled ("noise“) words instead of all possible words. In this way, the number of classes and hence the problem size is largely reduced.

The language models of ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2018) build upon similar ideas. In contrast to word2vec, however, the surrogate task in these approaches is to predict a word given its context, instead of the other way round. While GPT is an unidirectional "left-to-right“ model, which can only see the previous tokens in the text, BERT and ELMo propose a bidirectional model, which is able to exploit the context in both directions. Another important difference between the approaches is the model architecture. ELMo employs recurrent (Long Short-Term Memory (LSTM)) layers, while GPT and BERT build on the Transformer model (Vaswani et al., 2017), which uses neither convolutions nor recurrent operations, but bases only on attention mechanisms. The attention based mechanism has been shown to be very effective for language modeling tasks. For GPT and BERT the model is pre-trained on unlabeled data and fine-tuned for specific tasks. These approaches show how minimal architectural changes can be sufficient for different tasks, in this way being able to transfer the model pre-trained on unlabeled data to a wide range of tasks.

2.3.2 Learning image representations

Probably one of the most prominent surrogate tasks in the image domain is to reconstruct the input (Hinton and Salakhutdinov, 2006; Kirby and

Sirovich, 1990; Vincent et al., 2008). For this task the goal is to reconstruct the input from a usually much lower dimensional representation derived from the (transformed) input. In this way the lower dimensional representation has to retain the crucial information of the input and represent it in an abstract manner. While the Principal Component Analysis (PCA) (Hotelling, 1933) is an example of a linear model implementing such a task, an autoencoder implements a similar task with a non-linear model (Vincent et al., 2008). For training an autoencoder the input is often transformed or distorted, respectively. In a prominent approach coined denoising autoencoder (Vincent et al., 2008), pixelwise noise is added to the input and the target is to reconstruct the clean image. The idea is that similar appearances have to be associated in the training data in order to discriminate the noise from the original signal. Transforming autoencoders (Hinton et al., 2011) on the other hand are trained to generate a transformed version of the input according to a given transformation, *e.g.*, an (x, y) offset or a small rotation. Nevertheless, it appears questionable whether the model really has to associate semantically similar images to solve the reconstruction task – as we also point out for the task of hand pose estimation in Chapter 4.

More recently, Doersch et al. (2015) introduced a prominent attempt to transfer the ideas of context prediction surrogate tasks to computer vision. In their work they sample a random patch from an image and a second patch from the neighborhood and aim to predict the relative location of the two patches. They formulate the task as a classification problem where the neighborhood is discretized into eight regions (top, bottom, left, right, top-left, top-right, bottom-left and bottom-right) and the goal is to predict the class label corresponding to the relative location from which the second patch was sampled. Figure 2.2 illustrates the construction of this surrogate task. By training a model using this surrogate objective they show that semantically similar images are often close together in the feature representation. Moreover, using this surrogate task as the pre-training objective they show that object detection performance can be improved. Nevertheless, the results for pre-training with ImageNet labels (Russakovsky et al., 2015) are still significantly better.

Noroozi and Favaro (2016) showed that posing the task slightly different can be more effective as a pre-training task. They proposed to learn to solve jigsaw puzzles. For this they divide an image region in 3×3 square crops and randomly permute the crops. The crops are input to a network, whose objective is to predict the correct order of the crops. Again they formulate this as a classification problem, where a number of different permutations

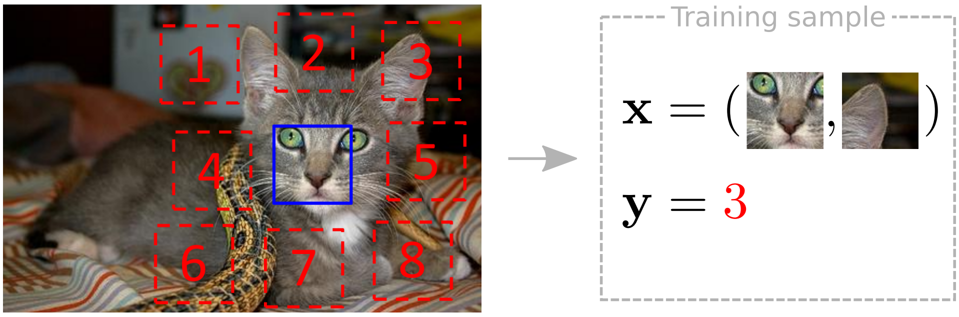


Figure 2.2: **Context prediction surrogate task.** Illustration of the formulation of context prediction as a classification task. A center and a surrounding patch are sampled and the model is trained to classify which of the predefined relative locations was sampled. Illustration adapted from Doersch et al. (2015).

are predefined and each permutation represents a different class. That is the objective is to predict the index of the sampled permutation.

Others formulate inpainting or colorization as surrogate tasks. For instance, the approach of Pathak et al. (2016) can be seen as a straight forward computer vision implementation of some of the language modeling surrogate tasks described above. In their work the surrogate task is formulated using inpainting. That is, they mask a region of the image and the task for the model is to predict the content of the missing region given the context of the image. Instead of masking parts of the image in the spatial dimension, other works reduce the number of image channels which are input to the model and, *e.g.*, propose the task of predicting color from grayscale images (Iizuka et al., 2016; Larsson et al., 2017; Zhang et al., 2016).

One crucial issue of the inpainting as well as color-prediction tasks is that during training the input is missing parts of the image or has a smaller number of channels, but at test time the full images are input. That is, the training samples are different from the test samples. To overcome this issue, a follow-up work (Zhang et al., 2017) proposes to split the input $\mathbf{x}$ into two disjoint parts $\mathbf{x}_1$ and $\mathbf{x}_2$ and train one model to predict $\mathbf{x}_2$ given $\mathbf{x}_1$ and a second model to predict $\mathbf{x}_1$ given $\mathbf{x}_2$. For instance, for color images in the *Lab* colorspace one model can be trained to predict a and b channels from the L channel and the other model should predict the L channel given a and b. In this way the concatenation of the two models can digest the full input, however, the learned representation is unable exploit correlations between the disjoint inputs since it cannot be trained jointly.

Other self-supervised approaches, which do not exhibit such a strong domain gap between training and test input, are based on, *e.g.*, image transformations. Dosovitskiy et al. (2014) for instance propose to generate surrogate classes by applying a number of image transformations to each sample. Each sample together with all its transformed variants makes up one class. Another approach relying on the correspondence between images of objects, which have undergone some transformations has been proposed by Wang and Gupta (2015). They used an off-the-shelf tracker to track moving patches within videos. Then the first and last frames of the tracked patches in some short sequences are used for learning a model. The learning objective is to move the representations of the related patches closer together than the representation of a random other patch using a triplet loss (Wang et al., 2014). More recently Gidaris et al. (2018) have proven another image transformation based objective to be very effective. In contrast to the aforementioned works they rely on the bias that most objects exhibit a natural upright direction and are captured accordingly by the photographer. They exploit this bias by inputing a rotated variant of an image to the model and requiring the model to predict the applied rotation. The intuition is that the model can only predict the correct rotation if it is aware of the objects and the relation of their parts.

Employing the existing self-supervised approaches for pre-training image representations have been shown to be more effective than training from scratch for datasets with a small number of labeled samples. The approaches have also been shown to match or even outperform results for pre-training with ImageNet labels for tasks, where the targets have no semantic meaning, like surface normal estimation (Doersch et al., 2015; Goyal et al., 2019). Nevertheless, – with the exception of some object detection benchmarks – pre-training with ImageNet labels still yields better results for semantic classification tasks (Goyal et al., 2019). Increasing problem complexity, *e.g.*, in the way of combining (some of) the proposed surrogate tasks, has been shown to be able to improve the results further (Doersch and Zisserman, 2017; Goyal et al., 2019). Another remaining issue, hampering research, are slower training convergence rates compared to ImageNet pre-training (Doersch and Zisserman, 2017; Goyal et al., 2019), which arise probably largely due to the weaker and less task specific supervision provided by the self-supervised tasks.

2.3.3 Learning video representations

In a captured scene, neighboring image regions are often naturally correlated, but also over time a scene exhibits natural temporal relations. Many works aim to obtain supervision from the temporally coherent processes captured by video sequences. Such a video sequence can be exploited by learning the temporal relations between activities – or more generally between any sequential procedures (Ostrovsky et al., 2009). For instance the natural temporal order of the frames exhibiting some motion can be exploited by *e.g.*, learning to predict future frames (Liang et al., 2019; Mathieu et al., 2016; Ranzato et al., 2014), verifying possible temporal sequences (Fernando et al., 2017; Misra et al., 2016), or tracking patches back and forth (Wang et al., 2019b). Conceptually, several of these approaches have a strong relation to the (uni-/bidirectional) language models discussed above, where the surrogate task is to predict the next or a missing token given the sequential context (Radford et al., 2018; Ranzato et al., 2014).

To learn to predict future frames from a video is a conceptually simple idea to make the model learn about natural temporal relations. Implementing the concept, however, exhibits significant obstacles. This task has been tackled, *e.g.*, by using LSTMs to generate a video representation from which the input sequence, *i.e.*, the past frames, can be reconstructed and at the same time the future frames can be predicted (Srivastava et al., 2015; Vyas et al., 2018). Other works focused on the difficulty to design a proper loss function for predicting such high dimensional targets under uncertainty and proposed, *e.g.*, to quantize the output space to transform the task to a less complex classification problem (Ranzato et al., 2014), or to employ an adversarial objective (Mathieu et al., 2016). The inherent ambiguities of the task and the – compared to language models – extremely high dimensional, continuous output space are, however, still hampering their application.

Instead of tackling the difficulties with predicting full frames reasonably far in the future, the prediction of abstracted information might represent a more feasible alternative. For instance, Liang et al. (2019) propose to predict future person and activity locations as surrogate tasks for predicting future activities. However, they assume that all person locations are known throughout the training videos, which itself can be difficult to achieve automatically. In another example Neumann et al. (2019) learn to predict, *e.g.*, if and when a car with the camera mounted on the windscreen stops. They show that such abstract events can be predicted and that the models learned to recognize abstract relations from such a task. For instance, the influence of the traffic lights or cars in front on a possible future stopping event were automatically discovered by the learned models.

Another way to have the model learn about natural temporal relations, but still circumvent the prediction of full frames, is to instead learn to verify a given sequence order, *i.e.*, verify if a given sequence appears natural. For instance, Misra et al. (2016) present the network with a sequence of frames and the objective is to decide whether the order is correct or not. That is, similar to NCE, for model training the positive samples are correctly ordered video sequences and the negative samples are generated by shuffling such sequences. Fernando et al. (2017) later pointed out that – instead of just concatenating the encodings of the individual frames as done by Misra et al. (2016) – using a sequence encoding and increasing the complexity of the task from binary to multi-class by having the network classify, which of several sequences has an incorrect order, can yield significantly stronger representations for action classification.

Nevertheless, a conceptual issue of these approaches is that the model get incorrectly ordered sequences as input during training, but will never see incorrectly ordered sequences during test time. This creates a domain gap similar to the one discussed above for works on learning image representations by inpainting (Pathak et al., 2016) or colorization (Larsson et al., 2017). In line with this, using the sequence verification pre-training task, the results for action recognition are stronger than when trained from a random initialization, but still significantly worse than for ImageNet pre-training. On the other hand, when training on videos of persons performing various activities, such a pre-training task has been shown to be very effective for pose estimation, where it can at least match the results of ImageNet pre-training (Misra et al., 2016).

While similar in spirit, the work of Wei et al. (2018) reduces the domain gap by simply playing the video sequences either forward or backward, and having the network learn to discriminate which direction it is. For this task – as for the related works discussed above – the selection of training samples is critical (Fernando et al., 2017; Misra et al., 2016; Wei et al., 2018). While for this task some cues, like gravity, are very strong, others, like constant motions, or motions which might easily appear in the opposite direction are unusable. Nevertheless, using this task to pre-train for action classification can outperform even ImageNet pre-training.

Other interesting areas for learning video representations are tracking applications. Very recently, self-supervised tasks have been proposed to learn tracking without any manual supervision. For instance, Vondrick et al. (2018) have shown that learning to colorize videos can be used as a valuable cue for learning video representations. The idea is based on the temporal coherency of color. In contrast to approaches to image colorization (Iizuka et al., 2016; Larsson et al., 2017; Zhang et al., 2016) they do not train the model to directly predict colors from a grayscale video, but they instead train the model to copy the color from some reference frame of the sequence. They argue that the model needs to learn to associate objects despite location changes or deformations and show that the model is able to track image patches when employing this surrogate task. Wang et al. (2019b) on the other hand obtain free supervision for learning to track from the observation that after tracking a patch forward and backward in time it should end up at the same location again. In their approach corresponding patches across frames are found by template-matching of the learned representation. The loss for learning the representation is based on the inconsistency between the start and the end point of a track. The authors showed that the model learned using this task can outperform optical flow based methods (Ilg et al., 2017; Liu et al., 2011) on applications like propagating instance masks, part labels or human pose keypoints. The model also matches the performance of training on ImageNet for propagating pose keypoints but is worse for other applications. Similar to the tasks discussed above, the selection of training samples, which should exhibit appropriate motion and reasonable appearance variation throughout the sequence, is again a crucial issue hampering performance.

2.3.4 Learning representations from multi-view data

In machine learning as well as computer vision literature the term multi-view is generally used to simply describe that different cues of information are exploited. When speaking about multiple views in this thesis, we refer to the multiple views provided by different camera view points. The different cues can, however, also be provided by different modalities, like text (Gomez-Bigorda et al., 2017; Gordo and Larlus, 2017), sound (de Sa, 1993; Owens et al., 2016) or any other signal (Dosovitskiy and Koltun, 2017; Tian et al., 2019). That is, one view would be provided by one modality, like a video camera, and another view would be derived from a different modality, like a microphone. For the case where the different views are provided by the same modality, they can also be generated from the same sensor, but from different locations in space or time (Li et al., 2018). When employing this definition many of the approaches, which were already discussed above, *e.g.*, where parts of a text, image or video are predicted from its context (Doersch et al., 2015; Mathieu et al., 2016; Mikolov et al., 2013b), can also be considered multi-view approaches. Therefore, after discussing some prominent general approaches, in this section, we subsequently focus our discussion specifically on works, where the multiple views are provided by multiple imaging modalities and especially on those which learn to predict one view from another.

Canonical Correlation Analysis (CCA) A very prominent strand of approaches for learning from multi-view data is based on CCA – a method introduced in the 1930th (Hotelling, 1936). CCA aims to find linear projections of two variables, which maximize the correlation between them. That is, with two corresponding datasets $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ representing two views of the data, the goal of CCA is to find two vectors

$\mathbf{a} \in \mathbb{R}^N$ and $\mathbf{b} \in \mathbb{R}^M$ such that the correlation between $\mathbf{a}^\top \mathbf{X}$ and $\mathbf{b}^\top \mathbf{Y}$ is maximized, *i.e.*,

$$(\mathbf{a}^\star, \mathbf{b}^\star) \triangleq \operatorname*{arg\,max}_{\mathbf{a},\mathbf{b}} \operatorname{corr}\left(\mathbf{a}^\top \mathbf{X}, \mathbf{b}^\top \mathbf{Y}\right), \tag{2.3}$$

where corr $(\cdot, \cdot)$ denotes the sample correlation of the two arguments. Later, also non-linear variants of CCA, like *Kernel CCA* (Kuss and Graepel, 2003; Lai and Fyfe, 2000) and *Deep CCA* (Andrew et al., 2013; Becker and Hinton, 1992; Wang et al., 2015) have been introduced and it has been extensively used for learning from multi-view data (Li et al., 2018; Sun, 2013).

CCA is one generic approach to multi-view learning. There are also many approaches, which specifically exploit multiple camera views for various tasks like detection, tracking or pose estimation (Poier et al., 2014; Simon et al., 2017). Often these works base on the dis-/agreement of the predictions from different views (Leistner et al., 2008; Rhodin et al., 2018b; Simon et al., 2017). Agreement between different views is also exploited in another prominent generic approach to multi-view learning, namely *co-training*.

Co-training With co-training (Blum and Mitchell, 1998), view specific models are iteratively trained and used to label unlabeled data for the other view. An example how to apply these ideas to exploit multiple camera views is provided by Leistner et al. (2008), which use ground-plane homographies to exchange detection results between the models for the individual camera views. Co-training enforces agreement between the models for each view and naturally only helps if there is (initial) disagreement between them (Krogel and Scheffer, 2004).

Other approaches to body, hand and object pose estimation exploit the fact that the locations estimated from different camera views should agree in 3D space. For example, assuming a known camera setup, such agreement can be enforced by training a Convolutional Neural Network (CNN) to directly output 3D coordinates, apply it to each camera view and transform the resulting 3D points according to the (known) camera poses in order to enforce agreement in the joint space (Rhodin et al., 2018b; Suwajanakorn et al., 2018; Wan et al., 2019). On the other hand, Pavlakos et al. (2017) employ a 3D pictorial structure model to find the 3D pose, which best explains the 2D keypoint heatmaps of each view. Other approaches triangulate the estimated 2D keypoint locations from different view points to obtain a 3D pose, which is then used to, *e.g.*, directly train a 3D pose estimator (Kocabas et al., 2019) or iteratively improve the 2D pose estimator by employing the back-projected triangulations as new ground truth (Simon et al., 2017).

Concurrently with our work in (Poier et al., 2018), other authors proposed to exploit multiple imaging modalities as supervision by learning to predict one from the other. For instance van Tulder and de Bruijne (2019) aim to learn cross-modal representations in the medical domain. They project the input from different modalities to different latent representations. The task proposed in their work is to reconstruct all input modalities from the averaged latent representations computed from (a subset of) the different modalities. The different modalities used in their work are different MRI sequences for the same subject. The goal is to produce a latent representation which is invariant to the different characteristics of, *e.g.*, different scanners. Similarly, Spurr et al. (2018) learn a cross-modal latent space for hand images (RGB or depth) and their poses. They show that using their approach unlabeled data can be exploited to improve pose estimation performance.

The idea of using different views for supervision by learning to predict one from the other has also been used in connection with actually different camera locations. The feedback which can be generated from a camera at a different location has, *e.g.*, been exploited for learning to predict depth for a given monocular image (Garg et al., 2016), or predict the second image of a stereo image pair to be used for 3D glasses or head-mounted VR (Xie et al., 2016). Especially the work of Garg et al. (2016) spurred a lot of follow-up works (Godard et al., 2017; Kuznietsov et al., 2017; Zhou et al., 2017a). Similar to Xie et al. (2016), they learn to predict depth/disparity, which is used together with given camera poses to warp the image to the second view. In this way a simple photometric error can be used to learn the model.

Similar ideas have also been used for tasks where the target is more explicit semantic, like the object instance, the pose or activities (Jayaraman et al., 2018; Rhodin et al., 2018a; Tatarchenko et al., 2016; Vyas et al., 2018). For instance, Tatarchenko et al. (2016) aim to reconstruct a full 3D model given only a single image of the object. They learn a model to reconstruct color and depth images from different views, which are subsequently used to compute a point cloud and a mesh, respectively. Their approach consists of an encoder and decoder neural network, where the decoder is provided with the desired camera pose of the output images as additional input. For training the model they had to rely on synthetic data, which was rendered from randomly sampled viewpoints. Very similar to the view prediction loss we employ in Chapter 4 and 5, Rhodin et al. (2018a) later proposed such an idea to learn a latent representation amenable to pose estimation. In their work they predict a latent representation consisting of 3D points, which are rotated according to the camera pose of the output view before being input to the decoder. Another approach following-up this strand was very recently proposed by Chen et al. (2019b), which – instead of trying to predict the full image of another camera view – aim to predict the 2D pose from the other view. Since their supervision is provided by 2D skeletons they avoid the necessity to capture appearance variations in the training set for 3D pose estimation. Instead they require an accurate 2D pose estimator to provide the supervision from the different views.

2.3.5 Learning representations for pose estimation

Finally we discuss some notable approaches to employ self-supervision specifically for pose estimation of articulated objects. For this we focus on approaches which have not been covered in any of the previous sections.

One strand of works on 3D pose estimation builds upon the fact that 2D annotations for related tasks, like keypoint detection or segmentation are much easier to obtain (Chen et al., 2019a; Pavllo et al., 2019; Wang et al., 2019a). For instance Chen et al. (2019a) propose to recover the 3D pose directly from the 2D joint positions, $\mathbf{y}^{(i)}$, estimated from an image. To learn a network which maps 2D poses to 3D without any 3D pose supervision they project the mapped 3D pose to a 2D pose $\hat{\mathbf{y}}^{(k)}$ in a virtual camera after applying some random 3D transformation $\mathbf{T}^{(k)}$. Subsequently, the same network is employed to map the 2D projection $\hat{\mathbf{y}}^{(k)}$ of the transformed pose to 3D again and the result is enforced to agree with the original (transformed) 3D pose. Similarly, they enforce agreement between the inversely transformed and back-projected 2D pose and the original 2D pose from the input image. Finally, they employ adversarial training to avoid degenerate solutions. To this end, a discriminator is trained to contrast the transformed and projected 2D poses with real 2D poses.

Another strand of works propose to learn to predict the parameters of a 3D model of the target object (*e.g.*, humans or hands) and build the training objective upon (differential) rendering of the 3D model. The rendering can then, *e.g.*, be contrasted with the observation in order to obtain a training loss for unlabeled data (Dibra et al., 2017). Nevertheless, several authors noted that methods, which learn to parametrize 3D models, are more difficult to train and more sensitive to errors in the estimation of parent nodes of the 3D model (Sun et al., 2017b; Wan et al., 2019; Zhou et al., 2016). In an attempt to overcome those issues Wan et al. (2019) recently proposed to approximate the hand surface with a set of spheres and directly estimate the sphere positions instead of the model parameters. In this way, the training can still exploit unlabeled data by rendering the spheres and minimizing the discrepancy between the rendered spheres and the depth image observation. To avoid degenerate solutions they rely on a learned pose prior, a loss term, which penalizes collisions, as well as constraints on the bone lengths.

Computing and rendering a realistic estimate of the target object to compare it with the observation is more difficult for the case when the input is a color image. Therefore some approaches combine the previous two ideas, *i.e.*, learn to estimate the parameters of a 3D model and enforce consistency of the reprojections on images with, *e.g.*, 2D joints or foreground masks (Baek et al., 2019; Boukhayma et al., 2019; Kanazawa et al., 2018; Tung et al., 2017). For example, Tung et al. (2017) learn to predict the parameters of a 3d body model by pre-training using synthetic data and subsequently use self-supervision losses to train on real data. The self-supervision is obtained from differential rendering of skeletal keypoints, dense motion, and foreground masks. The renderings are contrasted with ground truth or estimated 2D keypoints, estimated 2D optical flow and ground truth or estimated segmentation masks.

2.3.6 Discussion

From the literature review we see that self-supervised learning is attracting more and more research. Already from the mere number of works which have been published very recently we can see that the interest in self-supervision is strongly increasing. This is the case for a large range of target tasks, but especially for tasks where labels are difficult to obtain, like 3D pose estimation.

We also see that – partly due to the weak supervision – the selection of the surrogate task, which is used for self-supervised learning, is crucial for the performance on the target task. A surrogate task, which is very specific and as similar as possible to the target task is usually favorable for the performance on the ultimate target task.

2.4 Transfer learning

Another way to reduce the labeling effort for a given task is to exploit labels from a related task, for which more labeled data is available. Approaches, which follow this strand are usually subsumed under the term *transfer learning*. The goal of transfer learning can be described as improving the performance of a target task using the knowledge from other domains or tasks (Pan and Yang, 2010).

A source of labeled data we exploit in this thesis is synthetic data. By synthetic data we refer to data, which can be generated automatically –

together with accurate annotations. This has the advantage that a virtually infinite amount of accurately labeled data can be generated. Unfortunately, synthetic data is usually not distributed identically as real data. That is, a so called *domain gap* exists.

In the following we first provide a broad overview over transfer learning approaches in general including the necessary definitions (Section 2.4.1) and subsequently focus on ideas, which are more closely related to our work. In particular, we discuss the related background for the work we describe in Chapter 5, in which we specifically exploit synthetic data.

2.4.1 Overview

The interest in techniques for transfer learning started to increase in the 1990s. This start is sometimes associated with a workshop on "Learning to Learn“ held at the Conference on Neural Information Processing Systems (NeurIPS) in 1995 (Pan and Yang, 2010). The research on this topic has been referred to by many different names, which are sometimes used interchangeably but sometimes point out a different focus. These names include *e.g. knowledge transfer*, *learning to learn*, *meta learning*, *multi-task learning* or *domain adaptation* (Pan and Yang, 2010; Thrun and Pratt, 1998).

The different names for transfer learning sometimes also stress that a different kind of knowledge is transferred. For example for document classification a source task for which we have many labeled samples might be a binary classification task, where documents should be classified as relevant or irrelevant for the current thesis. A possible target task then might be to classify documents into ten classes according to the research strand they are following. An example for a different kind of knowledge transfer, would be if the documents from the source and target domain are categorized into the same ten classes, but are written in a different language, *e.g.*, documents from the source domain are Greek, while the target documents are German.

To make the different kind of transfer learning categories clearer we first define what we refer to by *domain* and *task*, respectively. To this end we follow the definitions and categorizations from Pan and Yang (2010), which are still applicable and followed in more recent surveys (Csurka, 2017; Wang and Deng, 2018).

Definition of a domain A domain Π is defined by an N-dimensional feature space $\mathcal{X} \subseteq \mathbb{R}^N$ and the marginal probability distribution over the feature space $p(\mathbf{x})$, *i.e.*, $\Pi = (\mathcal{X}, p(\mathbf{x}))$, where $\mathbf{x}$ is a random variable (Pan and Yang, 2010). Hence, we say that there is a domain difference between a source domain $\Pi_S = (\mathcal{X}_S, p(\mathbf{x}_S))$ and a target domain $\Pi_T = (\mathcal{X}_T, p(\mathbf{x}_T))$

if the feature spaces or the marginal distributions are different. Examples for different feature spaces, $\mathcal{X}_S \neq \mathcal{X}_T$, are different languages as in the document classification example above or different sensor modalities. A computer vision related example for the task of human pose estimation is the case where the source domain is based on RGB images of persons moving in a room, and the target domain is based on the reflections of radio frequency signals from another room (Zhao et al., 2018). On the other hand, we have different marginal distributions, $p(\mathbf{x}_S) \neq p(\mathbf{x}_T)$, in the document classification example if, *e.g.*, documents from the source and target domain are about different topics. A computer vision example are source and target domain images, which exhibit different noise patterns, as we have in this work for synthetic and real images.

Definition of a task According to Pan and Yang (2010), given a domain $\Pi = (\mathcal{X}, p(\mathbf{x}))$, a task E is defined by an M-dimensional label space $\mathcal{Y} \subseteq \mathbb{R}^M$ and the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$, *i.e.*, $\mathrm{E} = (\mathcal{Y}, p(\mathbf{y}|\mathbf{x}))$, where $\mathbf{x}$ and $\mathbf{y}$ are random variables. That means that for two different tasks $\mathrm{E}_S \neq \mathrm{E}_T$, the label spaces or the conditional distributions are different. An example for different label spaces $\mathcal{Y}_S \neq \mathcal{Y}_T$ was given above for the document classification example, where the source task was a binary classification and the target a ten-way classification task. Different conditional distributions $p(\mathbf{y}_S|\mathbf{x}_S) \neq p(\mathbf{y}_T|\mathbf{x}_T)$ can be induced by different label biases. For example, if the target task is pose estimation of professional skiers during a race and the source task was to estimate the poses of first-time skiers.

Transfer learning categories

To give a better overview over transfer learning approaches we categorize them into (i) *inductive transfer learning*, (ii) *transductive transfer learning* and (iii) *unsupervised transfer learning* (Pan and Yang, 2010). By inductive transfer learning we refer to approaches for which the source and target tasks are different, $\mathrm{E}_S \neq \mathrm{E}_T$, – independent of their domains. For transductive transfer learning, on the other hand, the source and target tasks are the same, $\mathrm{E}_S = \mathrm{E}_T$, while their domains are different, $\Pi_S \neq \Pi_T$. Finally the less prominent category of unsupervised transfer learning – like inductive transfer learning – targets the case when the source and target tasks are different, $\mathrm{E}_S \neq \mathrm{E}_T$, but no labeled data is available for both tasks.

Inductive transfer learning For inductive transfer learning, where the source and target tasks are different, some labeled data for the target task is needed to induce a model for the target task (Pan and Yang, 2010). For example, if at least some labeled data is available for both, the source and target task, this is similar to the setting of multi-task learning (Caruana, 1997; Li et al., 2019; Parameswaran and Weinberger, 2010). In this thesis we propose to exploit unlabeled data using a self-supervised surrogate task, for which we have (”natural“) labels, together with some labeled data for the pose estimation target task. Hence, this idea is related to the inductive transfer learning category. But also many recent deep learning approaches, which build upon a pre-trained model, are related to this category, as it has been shown that features learned for a source task – like image classification on ImageNet – can be re-purposed to different tasks (Donahue et al., 2014; Girshick et al., 2014; Long et al., 2015a).

Transductive transfer learning With transductive transfer learning the tasks are the same, but the source and target domain are different. It is usually considered as a setting where a large amount of labeled data from the source domain and unlabeled data from the target domain is available (Morerio et al., 2018; Pan and Yang, 2010). This category also includes a recently very prominent strand, which is usually called *domain adaptation* (Csurka, 2017; Wang and Deng, 2018). Domain adaptation is considered as the setting where the feature spaces of the two domains are the same, $\mathcal{X}_S = \mathcal{X}_T$, but their marginal distributions differ: $p(\mathbf{x}_S) \neq p(\mathbf{x}_T)$. This situation is sometimes also referred to by *covariate shift*. Similar to the case for inductive transfer learning, many recent approaches in this category employ pre-trained deep architectures to build their methods upon them (Csurka, 2017). On the other hand, a very prominent strand of approaches employs deep learning architectures specifically designed for domain adaptation. In contrast to shallow approaches built upon deep features, or fine-tuning strategies, these approaches design already the networks with the goal to make the learned representations more amenable to the knowledge transfer. The approaches usually build upon a siamese architecture Bromley et al. (1993) with one stream for the source domain and another for the target domain (Tzeng et al., 2017). Besides the standard task-specific objective, the model is trained to reduce the difference between the feature distributions in some latent feature space. The latter objective can, *e.g.*, be based on a discrepancy loss (Long et al., 2015b; Tzeng et al., 2014) or an adverarial loss (Ganin et al., 2016; Tzeng et al., 2017). The method, which we introduce in Chapter 5 of this thesis is partly inspired by these works on adversarial domain adaptation. In particular, the intuition behind one of the loss terms we employ therein is essentially the same as in these works.

Unsupervised transfer learning If there are only unlabeled samples available for both, the source and the target domain, Pan and Yang (2010) speak about unsupervised transfer learning. For this case the source and target tasks are different but related. Such approaches might be used for unsupervised learning tasks like clustering or dimensionality reduction. For example Dai et al. (2008) aims to find a clustering for a small number of samples in the target domain by simultaneously clustering a large number of samples from the source domain and learning a shared feature space for the two domains.

In Chapter 5 we show a way to combine ideas related to inductive and transductive transfer learning in a multi-task framework. We do this in order to particularly exploit synthetic data for our task. Hence, in the following we discuss various ways in which synthetic data has been exploited for training when the actual application requires handling real data. That is, we analyze how the respective domain gap has been handled and provide the background for the approach to domain adaptation we employ.

2.4.2 Learning from synthetic data

It has been shown that synthetic data can be generated for a large range of applications. Besides simpler silhouette or depth images of hands (Rosales and Sclaroff, 2006; Tompson et al., 2014) this also includes tasks for which the input are, *e.g.*, color images (Tatarchenko et al., 2016), and captures more complex scenes (Krähenbühl, 2018; Richter et al., 2016; Varol et al., 2017). Hence, the domain gap between synthetic and real data has to be tackled in a broad range of scenarios.

For a task, for which appropriate synthetic data can be generated the probably simplest approach is to ignore the domain gap at all. This has been surprisingly successful for some tasks (Mayer et al., 2018; Varol et al., 2017; Zimmermann and Brox, 2017). For example, Tatarchenko et al. (2016) learned to reconstruct 3D models of objects from single images by training on synthetic data from the ShapeNet dataset (Savva et al., 2015). While being trained solely on synthetic data, they show that their approach delivers reasonable results for real data. This has been enabled by using an appropriate rendering procedure, overlaying the rendered objects for training on random real image backgrounds, and ensuring that the real test images show the object in a rather iconic pose (*i.e.*, with uncluttered background and without occlusions) in the center of the image. Nevertheless, the predictions for real images seem significantly degraded compared to the results on synthetic data. Similarly, training only on synthetic data for our task yields insufficient results in many cases. The results of such an approach can easily be improved as we show in Chapter 5.

Instead of using the, usually very clean, rendered synthetic images directly, the rendered images can be distorted in order to better reflect the appearance of real data. In the simplest case the distortions can be rather straightforward adaptions like adding noise or blur to the images (Tatarchenko et al., 2016). But there have been efforts to imitate the camera noise more accurately.

For example, Planche et al. (2017) simulate the whole capturing process of real depth cameras to create realistic depth data from 3D models. They not only aim to model the degradations related to the sensor like the lens distortion and sensor noise, but also external factors like material properties and motion.

Instead of hand engineering a simulation of the capturing process other approaches aim to learn a model to map from a rendered synthetic image to a more realistic one. Since it is difficult to obtain a large variety of renderings of virtual models together with accurately corresponding real images, such approaches aim to learn the model using unpaired images. For example, this has been approached using Generative Adversarial Networks (GANs) (Liu and Mian, 2017; Mueller et al., 2018; Shrivastava et al., 2017), which should ensure that the output of the mapping from a synthetic to a real image appears realistic according to a discriminator network. The disadvantages of such GAN-based approaches include difficulties to train them (Arjovsky and Bottou, 2017; Sungatullina et al., 2018), and also that popular GAN models used for this task (Mueller et al., 2018), like CycleGAN (Zhu et al., 2017), have been shown to encode hidden information in order to solve their adversarial task (Chu et al., 2017), which can hamper their applicability for domain adaptation.

Another viable approach to exploit synthetic data is to first estimate an abstract intermediate representation, which is more similar for synthetic and real data but still discriminative for the target task. Then a model can be learned just based on the intermediate representations and targets from synthetic data (Doersch and Zisserman, 2019; James et al., 2019). For 3D pose estimation such an intermediate representation can, *e.g.*, be 2D keypoints, from which 3D poses can still be estimated (Doersch and Zisserman, 2019). Such an approach is advantageous if it is easier to obtain the intermediate representations for real and synthetic data than the ultimate target representation and the domain gap is smaller at the level of these intermediate representations.

Instead of hand defining task specific intermediate representations, like 2D keypoints, these intermediate representations can also be obtained from an intermediate layer of a neural network. However, since there is no reason to assume that the domain gap is smaller at an intermediate network layer, a small domain gap has to be explicitly enforced at this layer (Massa et al., 2016; Rad et al., 2018b). For example Massa et al. (2016) aim to detect exemplars by learning from synthetic images. They exploit the features learned from synthetic data by learning to map the features of real images to the features of synthetic images. The approach, however, requires a large amount of corresponding real and synthetic data to enforce the small domain gap. Establishing the correspondence naturally requires all the real data samples to be labeled.

A way to overcome the requirement for corresponding samples is related to generic domain adaptation approaches discussed above (Ganin et al., 2016; Tzeng et al., 2014), which aim to align the feature distributions between the synthetic source and real target domain (Huang et al., 2018; Rad et al., 2018a; Wu et al., 2018). For example, Huang et al. (2018) propose such a CNN model for semantic segmentation. They enforce that the distribution of the activations for the real data matches the activation distribution of synthetic data. The matching is enforced using an adversarial loss, but instead of only matching the output distributions as in related domain adaptation works discussed above (Ganin et al., 2016; Tzeng et al., 2017), they rather aim to align the distributions at each layer.

2.4.3 Discussion

In this section we provided an overview over approaches towards transferring knowledge from different tasks and domains in order to reduce the labeling effort for the target task. We especially focused on research directions related to the work we describe in Chapter 5, in which we aim to overcome the domain gap between synthetic and real data.

We have shown that a broad range of approaches has been proposed. Especially, recently the interest in this topic increased. Besides the generally increasing interest in many machine learning related topics, this can be related to a number of other reasons. These reasons include, *e.g.*, the advances in modeling and rendering realistic scenes (Krähenbühl, 2018; Richter et al., 2016), which enable the generation of increasingly realistic synthetic data, and that features pre-trained using deep architectures can be re-purposed for a surprisingly broad range of domains and tasks (Donahue et al., 2014).

Nevertheless, we have pointed out that many of the proposed approaches are insufficient for our task or exhibit (severe) drawbacks towards our goals. Instead of following a single specific approach, we draw inspiration from several of the approaches discussed in this section. Using these ideas, in

Chapter 5 we show that a large amount of unlabeled samples together with a rather small amount of labeled samples from the target domain can be used to significantly improve the results of related approaches.

2.5 Exploiting hand models

Previously in this chapter, we mainly discussed data-driven approaches. These data-driven approaches aim to learn a model from data, which can then be used to map from the input observation to the output pose.

We thereby ignored another possibility to estimate the pose of an object. Instead of having the system learn about the target task and object from data, we can encode our knowledge about this task and object and guide the system by this prior knowledge. In general, such prior knowledge has not only been shown to be crucial for humans to solve a novel task (Dubey et al., 2018; Spelke and Kinzler, 2007) but has also been shown to be effective for many computer vision tasks (Fischler and Elschlager, 1973; Koller et al., 1993; Kolmogorov and Zabih, 2004).

The probably most rigorous way to exploit prior knowledge for pose estimation is to create a graphical model of the target object and find the pose, which best fits the observation by means of analysis-by-synthesis. In this section we discuss the main approaches following the analysis-by-synthesis strand. This includes a discussion of the graphical hand models, which have been introduced for this case (Section 2.5.1), together with the strategies to find the best model-fit (Section 2.5.2). Additionally, we will discuss ways how such analysis-by-synthesis approaches have been combined with data-driven approaches aiming to exploit their complementary advantages (Section 2.5.3).

Encoded prior knowledge is usually more effective, the more specific the knowledge is to the task. That is, the discussion in this section will mainly focus on the task of hand pose estimation. Nevertheless, at least in terms of the general ideas there are many similarities to other tasks with the goal to estimate the pose of an articulated object, like human pose estimation. Hence, we will sometimes also refer to works on related tasks.

2.5.1 Hand modeling

For discussing the hand models, which have been employed for hand pose estimation, we start with a brief look at human hand anatomy. This does not only enable us to understand the origins but also the deficiencies of the individual models, which can eventually lead to advantages and limitations

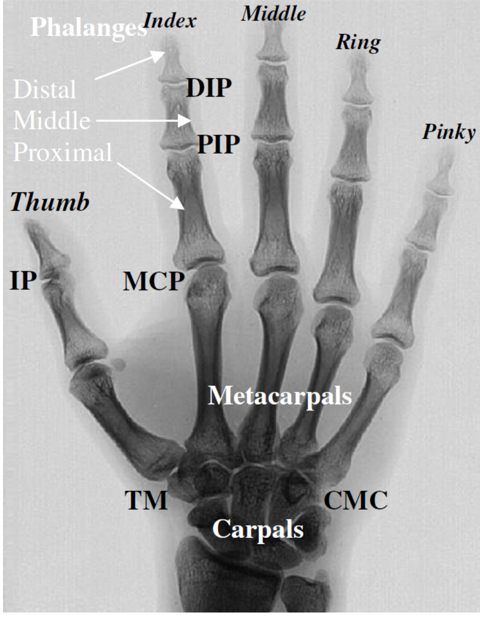


Figure 2.3: **The skeleton of the human hand.** An annotated illustration of the hand anatomy from Erol et al. (2007). The joint names are based on the names of the bones they connect: The interphalangeal joints (*IP) connect the finger bones, the metacarpophalangeal joints (MCP) connect the fingers with the metacarpals of the palm, and the carpometacarpal (CMC) connect the metacarpals with the carpals of the wrist. The CMC of the thumb is also called trapeziometacarpal (TM).

of the respective approaches. This will also allow us to make more informed decisions about the hand model for specific applications.

An example for the human hand and its bones is visualized in Figure 2.3. Usually, the human hand consists of 27 bones. The fingers and thumb are made up from the 14 phalanges. The five metacarpals connect the fingers with the wrist. And the wrist itself consists of the remaining eight bones (Erol et al., 2007)[2]. The bones are articulated by a number of muscles (usually around 30 to 40 (Sridhar, 2016)[3]), which, together with other soft tissue and the finger nails, define the final shape of the hand.

When aiming to model the pose of the hand in isolation, we have to consider the articulation capabilities of the respective joints connecting the bones. The articulations of the interphalangeal joints (*IPs) can be accurately defined with a single degree of freedom (DoF) for each of the nine joints. The metacarpophalangeal joints (MCPs) are often modeled with two DoFs. While using two DoFs for the MCPs is a reasonable approximation, it is also slightly restrictive – especially if the limited articulation capability

[2] See also: `https://en.wikipedia.org/wiki/Hand` (accessed: 2019-07-25)

[3] See also: `http://www.eatonhand.com/hw/facts.htm` (accessed: 2019-07-25)

of the carpometacarpal joints (CMCs) are not modeled either. In particular, the CMCs of the pinky and ring finger can be slightly articulated, which would lead to a arching or curving of the palm (Erol et al., 2007). The CMCs of the index and middle finger are rather static. However, the trapeziometacarpal joint (TM), *i.e.*, the CMC of the thumb, has two rotation axes, which are neither orthogonal to each other, nor to the bones. Additionally, the rotation axes are not intersecting each other (Hollister et al., 1992).

Model based approaches to hand pose estimation typically use about 10 to 50 DoFs to model the pose (Lin et al., 2000; Tompson et al., 2014). Most works model the global position and orientation of the hand with the full 6 DoFs and the articulation of each IP with a single DoF. However, since the CMCs articulation capabilities are rather difficult to model, many works differ in how they handle these difficulties. Some researchers assume the palm to be rigid (Lee and Kunii, 1993), ignoring the limited articulation capabilities of the CMCs. Others use three DoFs to model the MCPs of the fingers and the TM (Albrecht et al., 2003), or add a single DoF to specifically model the arching of the palm (Xu and Cheng, 2013). However, it has also been shown that the articulations of the hand exhibit much more constraints and can thus be represented by much less DoFs. Such constraints are imposed by, *e.g.*, the dependencies between some of the fingers and joints – like the bending of the ring finger caused by bending the pinky finger. Some researchers have shown that about 10 DoFs can be sufficient to achieve comparable accuracy for hand pose estimation (Douvantzis et al., 2013; Lin et al., 2000).

The defined skeleton models for the pose describe a hierarchy of rigid transforms. Each of these transforms essentially represents a joint of the skeleton. This hierarchy of transforms provides a set of constraints for the hand pose. For example, during test time the DoFs, which are estimated are the global position and orientation of the hand and the joint angles, whereas the distances between the joints are fixed. Besides such models, which employ hard constraints, researchers have also developed models, which rather impose soft constraints on the final pose (Donner et al., 2013; Sudderth et al., 2004). In such approaches the soft constraints are imposed by a prior model, which penalizes if neighboring parts are in some unlikely configuration, *e.g.*, too far apart or interpenetrating each other. However, in the following we focus our discussion on models employing hard constraints since they are much more frequently used and we employ such a model in Chapter 3.

The multitude of hand models, which have been introduced for the task of hand pose estimation, are used to find the pose by comparing them with

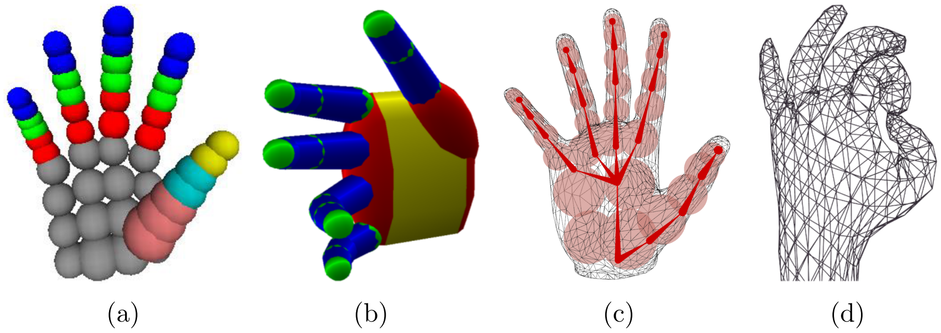


Figure 2.4: **Example hand models.** Approximation of the hand using (a) spheres (Qian et al., 2014), (b) cylinders and spheres (Oikonomidis et al., 2011a), (c) mixtures of Gaussians (Sridhar, 2016), and (d) using a full mesh (Taylor et al., 2016). Illustrations taken from the respective works.

the actual observation from the camera. Hence, a hand model need to be designed in a way that features can be computed from the model which are comparable with the (features of the) observation. Besides the minimum requirement that such features are computable, there are a number of other considerations which guide the model design. For instance the requirement to achieve real-time capabilities on a target device, or the interplay with different optimization algorithms *e.g.* requiring differentiability of the computed features, are often considered. Naturally, the different design decisions yield hand models with individual advantages and disadvantages. In the following we provide an overview of the employed models.

We show illustrations for examples of prominent hand models in Figure 2.4. The models usually present a trade-off between speed and accuracy with respect to the target application. Hence, some approaches build their models based on simple geometric primitives like cylinders or spheres (Oikonomidis et al., 2010; Rehg and Kanade, 1994; Stenger et al., 2001). Others construct a simple differentiable hand model based on, *e.g.*, a mixture of Gaussians (Sridhar et al., 2013). On the other hand, also more accurate models, which represent the hand surface as a 3D mesh have been constructed (Ballan et al., 2012; Schröder et al., 2014; Taylor et al., 2016). Some researchers even employ fully textured meshes to exploit cues provided by the shading (de La Gorce et al., 2011).

2.5.2 Model fitting

To make use of the prior knowledge encoded by a hand model, given an input observation, one usually wants to find the hand model parameterization, which best fits the observation. This parameter search has been conducted using a variety of optimization procedures. To get a clearer picture and understand the advantages and disadvantages of the respective approaches, we provide an overview of those approaches.

The approaches to hand model parameter optimization can be roughly categorized into two strands by considering whether they perform a *local* or a *non-local* parameter search. Due to the comparably high dimensional parameter space and that probably any effective objective function will exhibit a large number of local optima, an exhaustive global search is infeasible. Hence, the approaches, which we call *local*, assume that the initialization is close to, or at least within the basin of attraction of the optimal solution. On the other hand, the *non-local* approaches aim to escape local optima by using stochastic optimization procedures which can find optima outside the basin of attraction of the initialization.

Local methods often make use of gradient-based methods to quickly find an optimum (de La Gorce et al., 2011; Schmidt et al., 2015; Tagliasacchi et al., 2015). Hence, they employ differential functions to compare a model parameterization with the observation and obtain a search direction to find a better fit. That is, the hand model as well as the objective function need to be differentiable with respect to the model parameters. By computing the search direction based on the surface of the objective function they are able to quickly find a optimum. However, such approaches naturally encounter problems if the initialization is only within the basin of attraction of a bad local optimum.

Other methods do not only rely on the local surface of the objective function, but rather aim to explore a larger space. Since they are still bound by the computational budget, which makes a global search infeasible, they usually rely on stochasticity to find a good solution (Kyriazis and Argyros, 2013; Oikonomidis et al., 2010). That is, often they perturb the hand model parameters around one or several initializations and aim to find increasingly better parameters, *e.g.*, using an evolutionary strategy like Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995). In this way they can find a good solution even if the initial parameters lie in the basin of attraction of a bad local optimum. However, the final solution might not even be a local optimum since the local behavior of objective function is ignored.

Finally, the ideas from local and non-local approaches might also be combined. A number of opportunities for combination have already been proposed. For example, such a strategy might start with a stochastic optimization method and use the best found solution as initialization for a gradient-based optimization (Tompson et al., 2014), iteratively narrow the search space for a stochastic procedure over the course of optimization (Oikonomidis et al., 2014), employ a number of different random initializations (restarts) for a gradient-based optimization procedure (Taylor et al., 2016), or alternate stochastic and gradient-based updates (Qian et al., 2014).

Objective function The choice of the objective function is crucially connected with the choice of the optimization strategy. For example, gradient-based optimization procedures naturally require the objective to be differentiable, whereas gradient-free approaches are more flexible in their choice. In general, an objective function e computes the discrepancy or similarity v between features computed from the model and the observation:

$$v = e\left(\mathbf{z}_m, \mathbf{z}_o\right), \tag{2.4}$$

where $\mathbf{z}_m$ are features computed from the model and $\mathbf{z}_o$ are features computed from the observation. That is, comparable features, $\mathbf{z}_m$ and $\mathbf{z}_o$ need to be computable from the model and the observation. Often the model is rendered to derive features, which are directly comparable with the input, like the depth map (Oikonomidis et al., 2011a) or the color image (de La Gorce et al., 2011). Other works directly compute the distance between the input point cloud and the respective closest points on the hand model surface (Qian et al., 2014; Tagliasacchi et al., 2015; Taylor et al., 2016). Due to the large computational expense and the difficulties to realistically render the model many works also rely on (combinations of) intermediate representations, which can be derived from the model as well as the input, like image gradients (Rehg and Kanade, 1994), silhouette masks (Oikonomidis et al., 2011b), or joint positions (Tung et al., 2017).

2.5.3 Hybrid approaches

Despite elaborate strategies, model-based procedures have to rely on a good initialization. This is due to the highly articulated object, which causes vast appearance variations and a large parameter space with many local optima. That is, purely model-based approaches have to rely on an initialization in the first frame and are subsequently employed in a tracking framework, where the initialization for the optimization at every frame can be derived from the solutions at previous frames. Such a tracking framework naturally assumes constraints on the amount of movement between frames.

If the initialization is not close enough to the true pose the tracking can fail. This is especially problematic as model-based approaches are unable to recover from such tracking failures by themselves. We want to underline the severity of this issue with a theoretical consideration: For this, lets assume a tracking approach, which finds the correct pose in 99.9% of the frames. At a typical frame rate, for such an approach the probability to fail within 30 seconds is already significantly higher than 50%. That is, despite the accuracy of the tracking approach, such performance would clearly be insufficient for many real world applications like user interfaces for human computer interaction in many work or leisure related scenarios.

To overcome this issue, many works have proposed hybrid approaches combining a model-based strategy with a – usually data-driven – part, which is able to (re-)initialize the tracking (Rosales and Sclaroff, 2006). These hybrid approaches essentially follow one of two general strategies: Either they employ a model-based tracking strategy and only resort to a data-driven part for failure cases, or they follow the more prominent approach of employing a data-driven part to provide guidance for the model-based part at every frame.

For example, Wei et al. (2012) employ a model-based tracker for 3D human pose estimation and use a data-driven approach for initialization at the first frame and in failure cases. They aim to detect failure cases based on thresholding the discrepancies between the synthesized depth and silhouette images and the actual depth and foreground mask from the observation. For initialization at the first frame and recovery from failures, they train a Random Forest (RF) for pixelwise labeling, whose output is then used to guide the model-based optimization. Similarly, Ganapathi et al. (2010) employ body part detections and consider all parts, which are not already explained by the preceding model-based optimization step, in another refinement step of the model-based optimization. In contrast to Wei et al. (2012) they employ the part detections for error correction at every frame.

A more prominent strand of works use the data-driven part at each frame to initialize or guide the model-based part. For example, several works use some semantic information derived from the image, like detected keypoints (Ballan et al., 2012; Tzionas et al., 2016) or pixelwise part labels (Krejov et al., 2017; Roditakis and Argyros, 2015; Sridhar et al., 2015) to guide the model-based optimization. Qian et al. (2014), on the other hand, rely on heuristics to detect the finger tips, which are then used to find an initial hand model parameterization to start the optimization from. Other works propose to train a data-driven model to directly estimate the parameters of a hand model (Boukhayma et al., 2019; Dibra et al., 2017; Zhou et al., 2016), which can then be straightforwardly used to initialize the model-based optimization. Some approaches even use the data-driven part to provide a number of different initializations in order to overcome ambiguities and deficiencies of the learned single-frame model (Sharp et al., 2015; Taylor et al., 2016).

Beside the discussed approaches combining model-based with data-driven approaches for inference, there have also been efforts towards a stronger integration of the two parts during training. That is, for such approaches the graphical model is employed to guide the training of the data-driven part. In general, the intuition is that in this way the data-driven approach can focus more of its complexity to the cases, which would be difficult to solve for the model-based approach alone, but focus less on cases where the model-based approach alone will likely be sufficient (Boukhayma et al., 2019; Dibra et al., 2017; Ranftl and Pock, 2014). For instance, one such approach has been proposed by Dibra et al. (2017). They propose to train a CNN to directly predict the parameters of a hand model. During the training process not only the parameters of the CNN are updated, but also the hand model is refined. Dibra et al. (2017) focus on adapting the shape of the hand model during training and thereby fitting the hand model to a single user. In their work the supervision is only obtained from the hand model and comparing it to the observation. The idea of obtaining training-supervision in this way – instead of relying on labeled data – is closely related to ideas for self-supervised learning. Hence, a discussion of more works towards this end can be found in the respective section about pose estimation above (Section 2.3.5).

2.5.4 Discussion

In this section we discussed the exploitation of prior knowledge in related work. We have seen that representing prior knowledge using a graphical model and employing it in an analysis-by-synthesis approach provides an opportunity to approach the pose estimation task without the need for any labeled training samples. We discussed critical issues for such an approach – like the inability for effective (re-)initialization – when applying it in an isolated manner. Finally, we have also pointed out ways to overcome those issues. In particular, we have discussed various ways in which model-based approaches can be combined with data-driven approaches to account for the limitations of the individual approaches.

CHAPTER 3

Exploiting prior knowledge

Contents

Learning data-driven models for hand pose estimation usually requires a tremendous amount of labeled training samples. This hampers progress since providing enough samples is a large effort. In the previous chapter we noted that hybrid approaches, *i.e.*, combinations of data-driven and model-based parts, can help to overcome some of their individual issues. We investigate whether this also holds for the issues arising when a smaller amount of training data is available to train the data-driven part. Intuitively, a reduced amount of training data increases the uncertainty in the predictions of the data-driven part. In this chapter we introduce a hybrid method which especially exploits the uncertainties of the data-driven part to improve the results despite a smaller amount of labeled data. The chapter is largely based on a previous publication (Poier et al., 2015) in which we introduced the hybrid approach.

3.1 Motivation for a hybrid method

Most current approaches to tracking of hand articulations can be roughly categorized into model-based and data-driven schemes. In model-based schemes (de La Gorce et al., 2011; Melax et al., 2013; Oikonomidis et al., 2010; Wu et al., 2001) an underlying 3D hand model is used to render pose hypotheses, which are subsequently compared to the observations retrieved from the sensor. Since it is infeasible to search the whole range of possible hand poses, these methods rely on an initialization which already needs to be close to the true solution. Typically, the solution from the previous frame is used for initialization, which leads to problems in the case of very fast hand movements or dropped frames. Hence, subsequent tracking failures are hard to recover from.

On the other hand, data-driven schemes learn the mapping from specific appearances to hand poses from training data (Keskin et al., 2012; Tang et al., 2014; Xu et al., 2015). During testing, they usually infer joint locations independently from each other. In this way, the complex dependencies do not need to be modeled. However, the results are not constrained by hand anatomy or physics. Thus, the obtained pose estimates might be wrong or even impossible. Another issue for these approaches is that employing enough training data to densely cover the whole pose space is infeasible, because of the highly articulated nature of the human hand and the fact that the space of possible hand poses grows exponentially with the number of joints.

In this chapter we introduce a hybrid method with both data-driven and model-based elements that inherits the advantages of both paradigms. To this end, we follow the strand of hybrid approaches, which obtain (an) initial pose(s) based on the data-driven approach, and then validate and/or locally optimize the pose(s) using a model-based approach. This has been inspired in parts by several relevant and successful approaches on human pose estimation (Baak et al., 2011; Taylor et al., 2012; Ye et al., 2011). Nevertheless, for the task of 3D hand pose estimation, inherent difficulties like the substantial similarities between individual fingers and the very fast movements or complex finger interactions cause ambiguities and uncertainties which are often disregarded by previous works.

For deriving more informed decisions under uncertainty, Graphical Models have been proven very effective (Felzenszwalb and Huttenlocher, 2005; Fischler and Elschlager, 1973; Koller and Friedman, 2009). Hence, they have been extensively used in computer vision literature (Felzenszwalb et al., 2010; Geman and Geman, 1984; He et al., 2004). Besides being applied in a dense manner (Fulkerson et al., 2009; He et al., 2004; Krähenbühl and Koltun, 2011), sparse graphical models have become increasingly popular for tasks like object detection or pose estimation (Dantone et al., 2014; Felzenszwalb et al., 2010; Hamer et al., 2009; Wu et al., 2001). Despite their effectiveness, a naive implementation would lack efficiency due to the complex interactions which need to be modeled. To this end, approximations of the underlying distributions have enabled efficient inference (Felzenszwalb et al., 2010; Krähenbühl and Koltun, 2011; Krähenbühl and Koltun, 2013).

To exploit the inherent uncertainties in the task of 3D hand pose estimation, we first need to capture them. For this, we adopt successful work on body pose estimation (Girshick et al., 2011; Shotton et al., 2013) to train a regressor that is able to generate a distribution of location proposals for each joint. We input this distribution to a subsequent optimization procedure.

Within the optimization procedure we can then exploit the uncertainties, which are implicitly captured by the proposal distribution. To do this efficiently we employ an approximation of the full distribution upon which a graphical model operates. The optimization procedure considers multiple entirely different solutions for the global pose configuration, capturing the uncertainty of preceding processing steps. In this way, the regressor does not need to be perfectly accurate on its own but should only deliver a set of likely joint positions, which are subsequently refined. This also attenuates the need for a complete training database densely covering the whole pose space. Additionally, the whole process operates in 3D and is thus able to infer correct joint locations even in the case of occlusions and missing depth information. Moreover, optimization not only exploits the uncertainties, but also finds an anatomically valid hand pose, similar to inverse kinematics (see Figure 3.1).

3.2 Related work

In previous attempts to apply hybrid approaches specifically to hand pose estimation, the joint proposals provided by the data-driven approach are often refined by adding penalties to anatomically implausible joint locations (Poudel et al., 2013; Tang et al., 2013). However, the obtained hand poses can still be invalid since the refinement performs only a selection of the most plausible joint proposals and/or refines only some of the provided proposals. Hence, the approaches fail if all proposals for a single joint are inaccurate (*e.g.*, due to occlusions), or if the proposals are uncertain for many of the joints.

In contrast, our method introduces new joint positions, which respect anatomic constraints and, simultaneously, best fit the joint positions proposed by the data-driven approach in a global manner. Moreover, the pro-

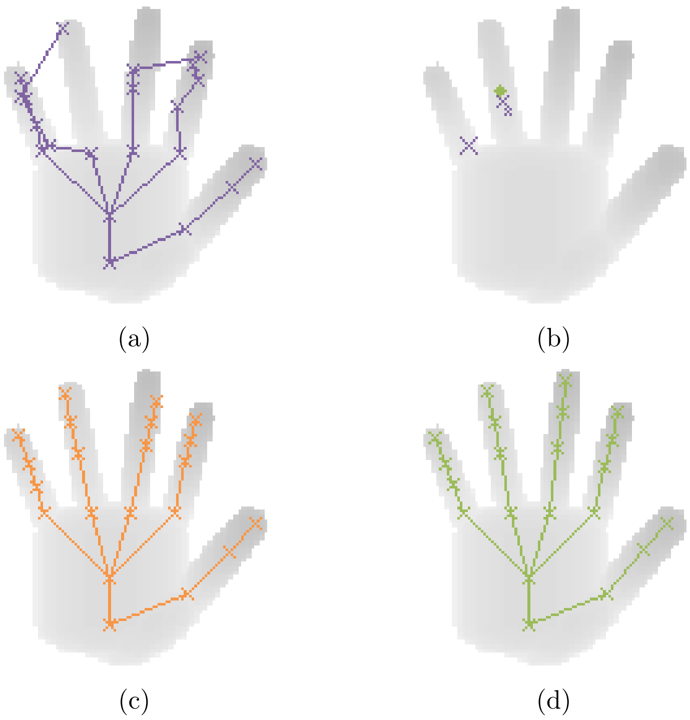

(a) (b) (c) (d)

Figure 3.1: **Overview.** (a) A learned joint regressor might fail to recover the pose of a hand due to ambiguities or lack of training data. (b) We make use of the inherent uncertainty of a regressor by enforcing it to generate multiple proposals. The crosses show the top three proposals for the proximal interphalangeal joint of the ring finger for which the corresponding ground truth position is drawn in green. The marker size of the proposals corresponds to the degree of confidence. (c) A subsequent model-based optimization procedure exploits these proposals to estimate the true pose. (d) The ground truth for this particular example. The same colors are used for the corresponding results throughout this chapter.

posed method does not rely on finding proposals of high confidence, but incorporates the approximated proposal distributions for each joint to find the anatomically valid hand pose which explains them best.

Another strand of research combines salient point detection with model-based optimization (Ballan et al., 2012; Qian et al., 2014; Tzionas et al., 2014). However, these approaches rely on detection of specific landmarks and will fail in situations where the landmarks (usually the finger tips or nails) are not clearly visible. In contrast, we do not rely on any landmark to be visible, but take a more holistic view considering the whole hand. Thus, our approach is robust to occlusion of specific landmarks.

Also noteworthy is an approach to 2D body pose estimation (Dantone et al., 2013, 2014). This Deformable Part Model (DPM) based work focuses on improving the unaries provided by a Random Forest (RF). In contrast to their work, we use hard constraints on the graphical model, which is enabled by defining it in 3D. However, their ideas for improving the unaries could potentially be applied to our task too.

Probably most closely related to our method is the approach of Tompson et al. (2014). This approach uses a deep Convolutional Neural Network (CNN) to infer the most likely positions of some predefined points on the hand which are then optimized by a model similar to ours. Despite employing depth information, regression is performed solely in 2D, disregarding occlusions or "holes" in the depth map. Moreover, in contrast to our approach, they rely on a single best location to fit the model, which ignores the uncertainty in the regression.

The basic building blocks of our method are similar to those of other hybrid approaches to hand pose estimation: a discriminative, data-driven method that generates likely joint positions (as in (Poudel et al., 2013; Tang et al., 2013)), and a generative, model-based optimization method that refines the initial solution (as in (Qian et al., 2014; Tompson et al., 2014)). However, the way we combine these two components is shown to outperform competing approaches. Another key difference that makes our method distinct from any other method we are aware of is that the optimization component has access to internal information of the data-driven component. It can thus make more informed decisions under the given uncertainty, which again yields significantly improved results.

3.3 Hybrid one-shot hand pose estimation

In this section we present the two building blocks which make up the proposed method. We use a discriminative regressor, which generates an approximation of the proposal distribution (Section 3.3.1). This distribution can be effectively transformed to anatomically valid pose hypotheses using the model-based optimization procedure described in Section 3.3.2.

3.3.1 Joint regression

For the generation of an approximated proposal distribution we build upon the prominent approach from Shotton et al. (2013). The approach relies on Random Forests (RFs) (Amit and Geman, 1994; Breiman, 2001; Criminisi et al., 2012) to infer a 3D distribution of likely hand joint locations. This approach has been shown to work well for real world applications of body pose estimation (Shotton et al., 2013), and has also been previously adapted for hand pose estimation (Tang et al., 2013). We briefly describe the training and testing procedures as applied in this work since its internal information is later exploited during optimization (Sec. 3.3.2). For more details the interested reader is referred to the related work.

Training For our task we follow a part based approach to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$. An input sample $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ represents the local appearance of an image patch around a foreground pixel from which we want to infer the 3D locations of J joints, *i.e.*, $\mathcal{Y} \subseteq \mathbb{R}^{3\times J}$. A training sample is formed by associating an image patch $\mathbf{x}$ with a part label c and corresponding offset vectors $\mathbf{o}$ pointing from the patch center location to each of the joint positions. Hence, the training set $\mathcal{L} = \{\Upsilon_i\}_{i\in\{1,\ldots,N\}}$ with $\Upsilon_i = (\mathbf{x}_i, c_i, \mathbf{o}_i) \in \mathcal{X} \times \mathcal{C} \times \mathcal{O}$, where $c_i \in \mathcal{C} = \{1, \ldots, J\}$ denotes the class label of the joint which is closest to the location of $\mathbf{x}_i$, and $\mathbf{o}_i \in \mathcal{O} \subseteq \mathbb{R}^{3\times J}$ denotes the set of 3D offset vectors. The training data is recursively split by each tree individually, until the maximum depth of a tree (23 in our case) is reached or less than a minimum number of samples (40) arrives at a node. For the experiments we fixed the number of trees to three.

Inspired by Schulter et al. (2011), we sub-sample the data arrived at a node for split selection. This not only speeds up the training process and enforces de-correlation between the trees, but also implicitly accounts for the different number of samples which are extracted per class by drawing a balanced sub-sample. The learned split functions are based upon the same simple depth features used in (Shotton et al., 2013). Finally, to generate the prediction models at the leaves, mean-shift (Comaniciu and Meer, 2002) is applied to the collected offset distributions for each joint (Shotton et al., 2013). The modes computed by mean-shift define the final offset vectors used at test time. Additionally, each mode, *i.e.*, offset vector is associated with a confidence based on the number of offset vectors, which ended up in the mode.

Evaluation During test time we start with an empty set of proposals $\mathcal{P}_j = \emptyset$ for each joint j. Image patches are sampled densely from the foreground region of the depth image and are passed down through each tree of the forest. Assuming that a test sample $\mathbf{x}$ arrives at leaf l_t of tree t, its 3D center position $\mathbf{c}_\mathbf{x}$ is offset by each of the offset vectors $\mathbf{o} \in \mathcal{O}_j^{(l_t)}$ for each joint j stored at the leaf to obtain proposals for the respective joint:

$$\mathbf{p} = \mathbf{c}_\mathbf{x} + \mathbf{o}, \quad \mathbf{o} \in \mathcal{O}_j^{(l_t)}. \tag{3.1}$$

In this way a set of proposals $\mathcal{P}_j^{(l_t)}$ is formed for each joint j, *i.e.*,

$$\mathcal{P}_j^{(l_t)} = \{\mathbf{p}_{j_r}\}_{r\in\left\{1,\ldots,\left|\mathcal{O}_j^{(l_t)}\right|\right\}}. \tag{3.2}$$

The set of offsets obtained from the associated leaf of each tree t is then added to the current set for joint j:

$$\mathcal{P}_j \leftarrow \mathcal{P}_j \cup \mathcal{P}_j^{(l_t)}, \quad \forall t \in \{1, \ldots, |\mathcal{T}|\}, \tag{3.3}$$

where $\mathcal{T}$ denotes the set of trees.

Following Shotton et al. (2013) we keep only a reduced set of top confident proposals $\tilde{\mathcal{P}}_j \subseteq \mathcal{P}_j$ for each joint and, subsequently, perform mean-shift on those to extract the k top modes $\hat{\mathcal{P}}_j$. We thus end up with at most $k \times J$ final proposals, each associated with a confidence score. The confidence score is based on the number of initial proposals supporting the mode. Here, we use this set of proposals and confidences as an approximation of the proposal distribution for each joint.

3.3.2 Model-based optimization

Using the discriminative RF based method described above, inference of the individual joint proposals is completely independent from the other joints. While, in this way, the complex dependencies do not need to be modeled, the resulting proposals are not necessarily compatible with anatomical constraints.

In order to obtain a valid pose we employ a predefined 3D model of a hand. We use a model with 26 degrees of freedom (DoFs). In this model the global pose of the hand, *i.e.*, position and orientation, has six DoFs, and each of the five fingers is specified by four more. These four parameters per finger encode angles where the base joint of each finger is assigned two DoFs and the two remaining hinge joints are each assigned one DoF. See Figure 3.2 for a skeleton visualization with the corresponding DoFs. During optimization these DoFs are constrained based on anatomical studies (Albrecht et al., 2003; Lin et al., 2000) which avoids impossible configurations. Since a quaternion representation is used for the global orientation, the 26 DoFs are modeled by 27 parameters.

While the used hand model is, in principle, similar to what is used in related work on hand pose estimation and tracking, our model only specifies the joint positions instead of specifically designed geometric primitives (Oikonomidis et al., 2011a; Qian et al., 2014), or even a complete mesh (Khamis et al., 2015; Sharp et al., 2015; Taylor et al., 2014). As pointed out later in this section, this has important implications on the computational complexity of the optimization process.

Having defined a hand model, the goal is to find the 27 model parameters which best describe the modes $\hat{\mathcal{P}}$ of the joint proposals, obtained from the regression forest. To this end, the objective function $e(\hat{\mathcal{P}}, \mathbf{h})$ judges the

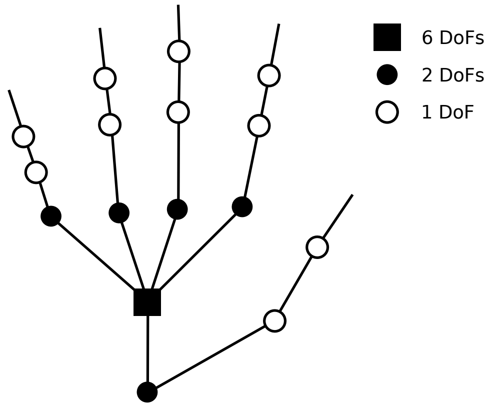


Figure 3.2: **Skeleton model.** Illustration of the skeleton model of the hand with the respective degrees of freedom (DoFs) of each joint used for model-based optimization.

quality of any hypothesized parameter set $\mathbf{h} \in \mathbb{R}^{27}$. More specifically, given a function $\delta_j(\mathbf{h})$ which extracts the position of joint j from hypothesis $\mathbf{h}$, the objective is formulated as:

$$e\left(\hat{\mathcal{P}}, \mathbf{h}\right) = \sum_{j=1}^{J} \max_{r} \left(w_{j_r}\left(1 - d_{j_r}^2\right)\right), \tag{3.4}$$

where

$$d_{j_r} = \min\left(1, \frac{\|\mathbf{p}_{j_r} - \delta_j(\mathbf{h})\|_2}{d_{max}}\right). \tag{3.5}$$

Here d_{max} is the clamping distance, $\mathbf{p}_{j_r} \in \hat{\mathcal{P}}$ denotes the r-th mode of proposals for joint j, and w_{j_r} is the normalized confidence so that it exhibits the properties of a probability. Intuitively, the objective enforces those modes to be "selected" which – together – best form an anatomically valid pose. The selection of modes, which contribute to the objective, is guided by the confidence, *i.e.*, the importance of each mode. Moreover, by considering all the top modes of the proposal distribution for a joint, we overcome the problem of outlier modes (*e.g.*, proposals for the wrong finger). The model will simply converge to joint positions close to those modes which best fit into the overall model. This is achieved by optimizing the objective for the best parameter set $\mathbf{h}^{\star}$:

$$\mathbf{h}^{\star} \triangleq \arg\max_{\mathbf{h}} e\left(\mathcal{P}^{(m)}, \mathbf{h}\right). \tag{3.6}$$

It is important to note that the definition of the objective function is based on a small number of 3D distances. As a result, no 3D hand model rendering is required to evaluate the objective function as, *e.g.* in (Oikonomidis et al., 2011a; Sharp et al., 2015). Hence, our objective function can be computed very efficiently on conventional (*i.e.*, CPU) processors.

For optimization of the objective function we follow other works on 3D hand pose estimation (Kyriazis and Argyros, 2013; Oikonomidis et al., 2011a; Qian et al., 2014; Sharp et al., 2015; Tompson et al., 2014) by employing Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995). PSO performs optimization by evolving a number of particles (solutions) that evolve in parallel over a number of generations (iterations). If not stated otherwise, an overall number of 50 generations turned out to be sufficient for our experiments. The incorporated randomness, which is introduced during the initialization of the particles as well as during their evolution, makes the method well suited for optimizing the non-convex, non-smooth objective function.

Stepwise optimization Any search space grows exponentially with the number of its dimensions. Hence, decomposing the space into non-overlapping sub-spaces offers the possibility of a large speed-up. We exploit this for hand pose estimation based on the observation that, given the global orientation of the hand, the fingers can move almost independently of each other. This independence together with the performed mapping of the proposals to specific joints allows us to split the optimization problem into sub-problems of lower dimensionality (*c.f.*, (Wu and Huang, 1999)). In line with this, we treat the problem of finding the best 27 parameters as six sub-problems, where we first optimize for the 7 parameters specifying the global pose of the palm, and subsequently individually optimize for the 4 parameters of each finger.

3.4 Experiments

We prove the applicability of the introduced method by means of several experiments on different datasets. A crucial requirement for benchmark datasets is the availability of ground truth annotations. However, accurate 3D annotations for real data are not easily obtained, especially for articulated self occluding objects like the human hand. To overcome this issue, we employ synthetic data in addition to real data. To generate a synthetic sequence which resembles natural movements, a hand is tracked with the method described in (Oikonomidis et al., 2011a) using the publicly available implementation[1] and with a very high computational budget. We then

[1] publicly available at `http://cvrlcode.ics.forth.gr/handtracking/`

render depth maps from the resulting poses and utilize the renderings as the test sequence. In this synthetic sequence the hand performs various finger articulations and typical motions like counting, pinching and grasping. While the performed movements are slow to ensure that the tracker does not get lost, we afterwards sampled every 5th frame from the sequence to simulate a more natural speed of movements. This sequence is referred to as *TrackSeq*.

For producing training data, we first defined 4 different articulations per finger. All 1024 combinations of these articulations were used as an initial set of poses. These poses were then rendered under 7 different viewpoints to create the full train set[2] of 7168 poses.

For experiments with real data we employ the *ICVL Hand Posture Dataset*[3] and *NYU Hand Pose Dataset*[4], where we use the available training and test data as is. The ICVL dataset was acquired using the Intel Creative *Time-of-Flight* (ToF) camera and includes a training set with roughly 330k images and two test sequences with 702 and 894 images, respectively. The two test sequences show a hand facing towards the camera performing various finger articulations in very fast succession. The NYU dataset was acquired using the Kinect RGB-D camera and includes a training set with roughly 73k images and a test set capturing two actors and consisting of 8252 images (2440 and 5812, resp.). However, neither our method, nor the approach of Tompson et al. (2014), who published the dataset, can yield meaningful results for the second actor. Table 3.1 shows the results on the full test set. For both methods the error and standard deviation is rather high, which is a consequence of having only the hand of a single actor in the training set, while the hand of the second actor in the test set differs significantly from the single hand in the training set. This is underlined by the significantly better results for the hand of the first actor (below 20 mm for our method) in the subsequent experiments. This brings us to the conclusion that both relevant methods for this chapter cannot really handle the large differences between the training and test set of this dataset. Hence, for a conclusive comparison in this chapter we compare only on the test sequence of the first actor for the NYU dataset.

[2] Note, that rendering from a different viewpoint is equivalent to changing the orientation of the whole hand.

[3] publicly available at `http://www.iis.ee.ic.ac.uk/~dtang/hand.html`

[4] publicly available at `http://cims.nyu.edu/~tompson/NYU_Hand_Pose_Dataset.htm`

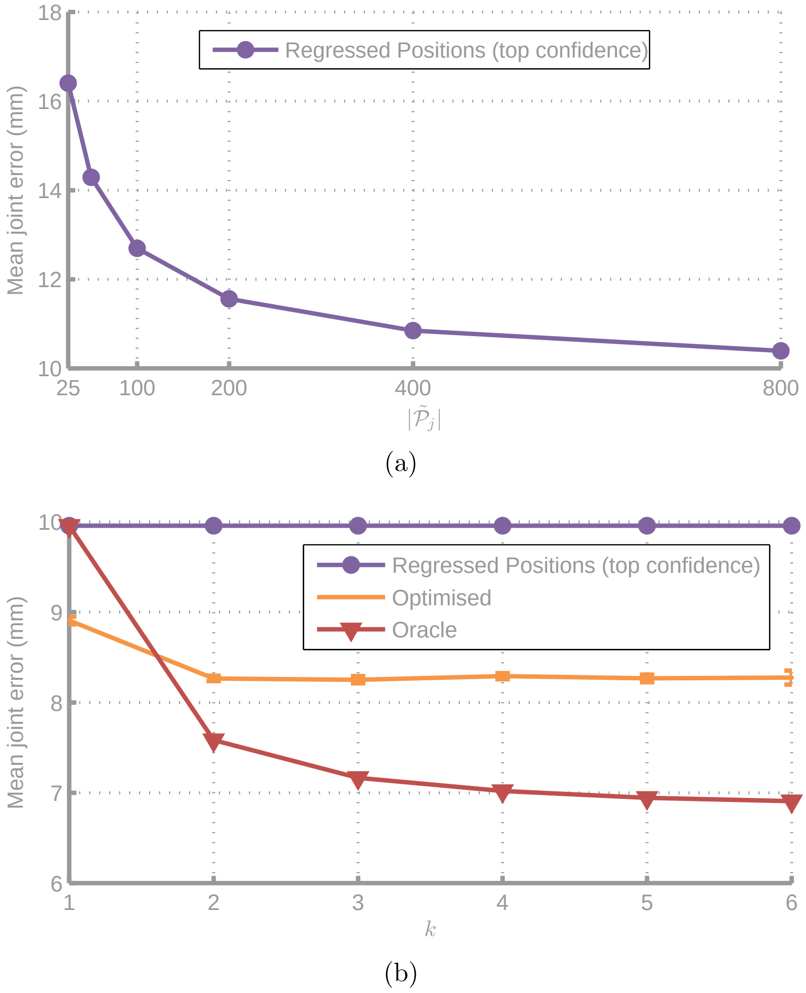
Table 3.1: **Results on full NYU test set.** Mean joint error (ME) and standard deviation over the errors for all joints of the original NYU test set.

	NYU ConvNet (Tompson et al., 2014)	Ours
ME (mm)	31.38 ± 82.95	31.67 ± 29.36

3.4.1 Influence of major processing steps

Size of proposal sets $\tilde{\mathcal{P}}_j$ An important parameter of our method is the number of proposals, $\left|\tilde{\mathcal{P}}_j\right|$, which are input to mean-shift at test time (see Sec. 3.3.1). This is especially interesting since mean-shift is responsible for summarizing the proposal distribution that we want to exploit during optimization. As can be seen from Figure 3.3a, the more proposals, which are used in this step, the lower the error. However, the error decrease becomes smaller for a higher number of proposals. Since a higher number of proposals implies higher runtime cost, the fact that the error levels off enables us to find a good trade-off between speed and accuracy. In our case, we fix $\left|\tilde{\mathcal{P}}_j\right| = 200$ for all other experiments. Another interesting observation is that accuracy seems to level off much later than reported for body pose estimation by Shotton et al. (2013). We hypothesize that this difference is due to the higher variation of hand poses compared to body poses within the respective datasets. In any case, it further advocates the specific consideration of the uncertainty as introduced in this chapter.

Number of final proposals per joint The optimizer can efficiently exploit the inherent uncertainty of the regression process due to the approximation of the proposal distribution using k modes of the distribution. In another set of experiments we investigate the effect of the number of generated proposals k on accuracy. Figure 3.3b shows the error with respect to k. The *Oracle* always selects the proposal closest to the respective ground truth joint position. Obviously, the more proposals generated, the closer one of them will be to the ground truth. The error for our method (*Optimised*) is higher because the solution has to respect the anatomical constraints of the hand. Interestingly, accuracy levels off for a small number of proposals (2-3) both for *Oracle* and for *Optimised*. Hence, the results show that utilizing a small number of proposals (together with confidences) instead of the full proposal distribution $\mathcal{P}^{(i)}$ is already very effective. For the other experiments we thus set $k = 3$. In fact this also limits the complexity of the objective function (Eq. (3.4)).



(a)

(b)

Figure 3.3: **Influence of meta parameters.** (a) Mean joint localization error on ICVL dataset as a function of the number of proposals, $\left|\tilde{\mathcal{P}}_j\right|$, which are input to mean-shift to extract the final joint proposals at test time. (b) Mean joint localization error on the *TrackSeq* sequence as a function of the number of top proposals, k. For *Optimised*, instead of markers, there are error bars showing the standard deviation over multiple runs.

Furthermore, Figure 3.3b indicates that the results of plain regression can already be improved by applying optimization to the single top proposal for each joint. This improvement can be attributed to the anatomically valid solution induced by model-based optimization. As also suggested by the results (for $k = 1$ and $k = 3$) in Figure 3.4, a significant additional gain is achieved by providing the optimization procedure with internal information about the uncertainty of the regressor, *i.e.*, multiple proposals per joint.

Stepwise optimization We investigate the effect of the stepwise optimization procedure described in Sec. 3.3.2. For a meaningful comparison we fix the overall number of objective function evaluations (*i.e.*, the optimization budget) for both approaches. We used 91 particles and generations when optimizing all parameters together, whereas for the stagewise optimization we assigned 64 particles and generations to the optimization of position and orientation and 29 particles and generations to optimization of each finger. The results are shown in Figure 3.4 (orange curves), from which we see that the stepwise optimization procedure proves much more effective.

Runtime Our current implementation of the Random Forest based regressor takes ∼33 msec on an Intel i7-4820K CPU to compute the joint proposals which are input to the optimizer. Optimization itself obviously depends on the budget. For the current implementation 1000 objective function evaluations take ∼10 msec. If not stated otherwise, we fixed the number of objective evaluations to ∼3400 for all experiments for a good speed vs. accuracy trade-off. Note that all timings are given for our current prototype implementation, which is non-optimized and *single* threaded.

3.4.2 Comparison to the state of the art

We compare our method to a state-of-the-art model-based (FORTH (Oikonomidis et al., 2011a)), data-driven (LRF (Tang et al., 2014)) and hybrid (*NYU ConvNet* (Tompson et al., 2014)) approach. Note that, all our results can be found on the project webpage[5].

Comparison with FORTH (Oikonomidis et al., 2011a) We conduct the comparison with the approach from FORTH on the *TrackSeq* test sequence. We do this since the hand model from FORTH was also used to generate this dataset and hence possible issues due a misfit between the hand model and the actual hand can be disregarded. Figure 3.4 compares the frame-based *success rate* over a number of distance thresholds. The frame-based success rate gives the ratio of frames in which all joints are estimated within a certain threshold to ground truth. The results show that our plain regressor performs similarly to the approach from Oikonomidis et al. (2011a) for the most interesting range of thresholds. However, after enforcing anatomic constraints by model-based optimization, the introduced hybrid method is able to improve on these results by a large margin. This is

[5] Results and other material can be found at `https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/hybridhpe/`

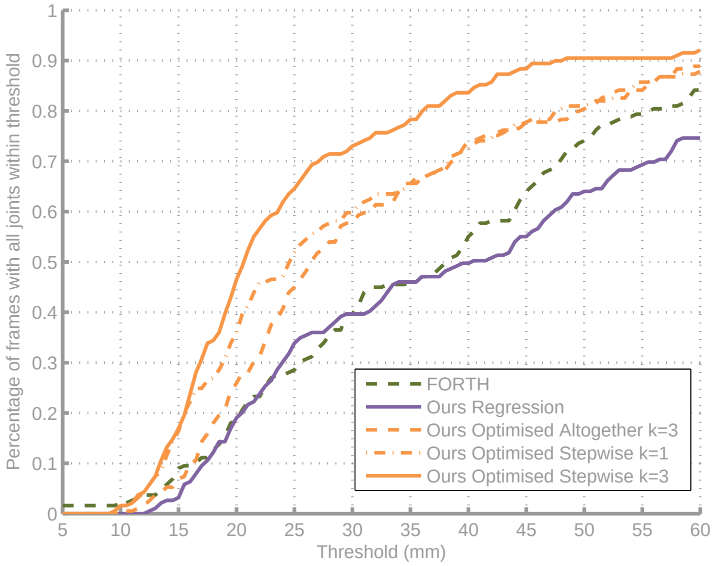


Figure 3.4: **Frame-based success rates on TrackSeq.** Results for different variants of our method and the approach from *FORTH* (Oikonomidis et al., 2011a).

achieved despite the fact that the sequence exhibits strong finger articulations, whereas the position and orientation do not change. Thus, the main reason for this gain is the improved estimation of finger articulations rather than the overall position and orientation estimation. This is also illustrated when solely considering the errors in finger tip localization, where the mean error for Oikonomidis et al. (2011a) is 19.5 mm, while for our approach it is 11.8 mm – an error reduction of about 40%.

Comparison with LRF (Tang et al., 2014) We compare to the Latent Regression Forest (LRF) approach on the ICVL dataset, which was published by the same authors (Tang et al., 2014). We compare to the results, which they published online. As shown in Figure 3.5 and 3.6 our regressor outperforms their results over most of the thresholds by a significant margin.

Unfortunately, we cannot fairly evaluate our model-based optimization using the annotations provided with the ICVL dataset. This is because in the ground truth annotations, bone lengths[6] vary significantly between the frames of a single sequence; therefore they are not compatible with an anatomically valid 3D hand. However, the 26 DoFs hand model used in this chapter implicitly applies strict constraints on them. Fitting such a model will therefore always introduce additional errors when compared to

[6] By bone lengths we refer to the distance between joint annotations connected by bones

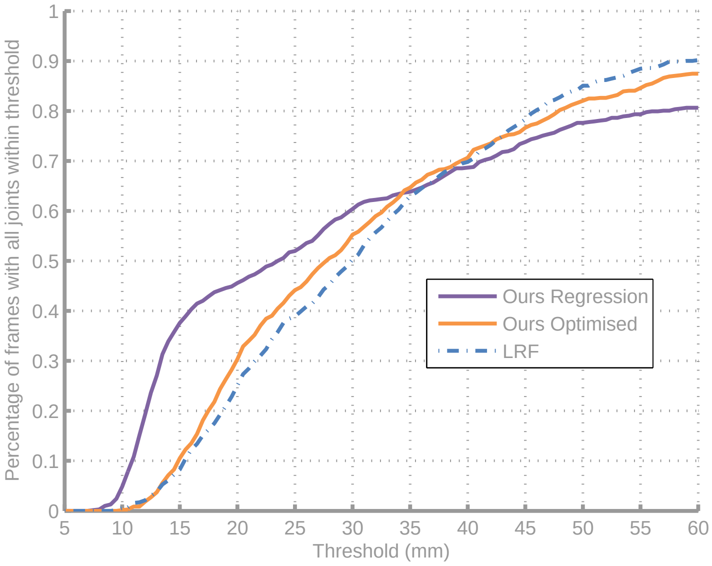


Figure 3.5: **Frame-based success rates on ICVL Sequence 1.** Results for variants of our method and *LRF* (Tang et al., 2014).

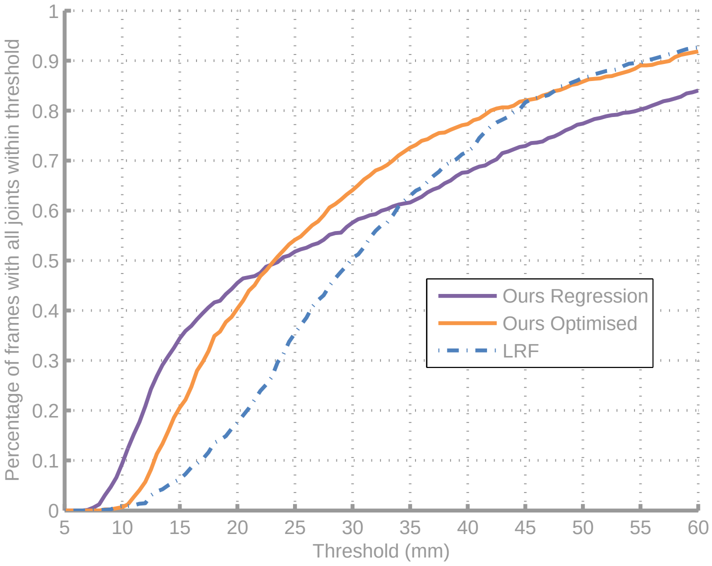


Figure 3.6: **Frame-based success rates on ICVL Sequence 2.** Results for variants of our method and *LRF* (Tang et al., 2014).

the provided ground truth annotations. Nevertheless, our results appear more accurate or at least as accurate as those from LRF (Tang et al., 2014).

Comparison with *NYU ConvNet* (Tompson et al., 2014) We compare to the hybrid approach from Tompson et al. (2014) based on the NYU dataset, which the same authors published along with their approach. For

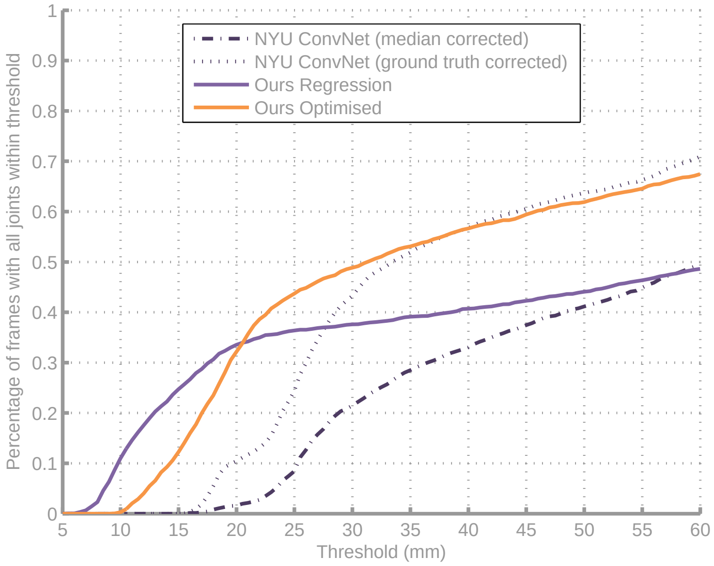


Figure 3.7: **Frame-based success rates on the NYU dataset.** Results for variants of our method and the hybrid approach from Tompson et al. (2014) (*NYU ConvNet*).

this dataset the annotated positions do not actually correspond to joints, but to some specific locations on the hand. For evaluation we used the suggested positions with the minor exception that we skipped two of the three palm positions since the palm is only represented by a single position in our model. In addition, for their approach Tompson et al. (2014) provide only 2D locations, which are projected to 3D using the input depth. However, the input depth might be significantly distorted (*e.g.*, at holes). To correct their estimates at positions with distorted depth, we augment them with the median depth of the inferred positions in the same frame and for a second variant with the ground truth depth to show the theoretical upper bound for their approach. Figure 3.7 and 3.8 show the results. For our method the optimization improves results particularly for larger distance thresholds since the optimization mainly performs an error correction rather than improving estimates which are already very close to the ground truth. In addition, the difference between the annotation model and our model induces an error, especially for low thresholds. In spite of that, we observe that our method outperforms the approach from Tompson et al. (2014) by a large margin – even if we correct their results by augmenting ground truth depth information.

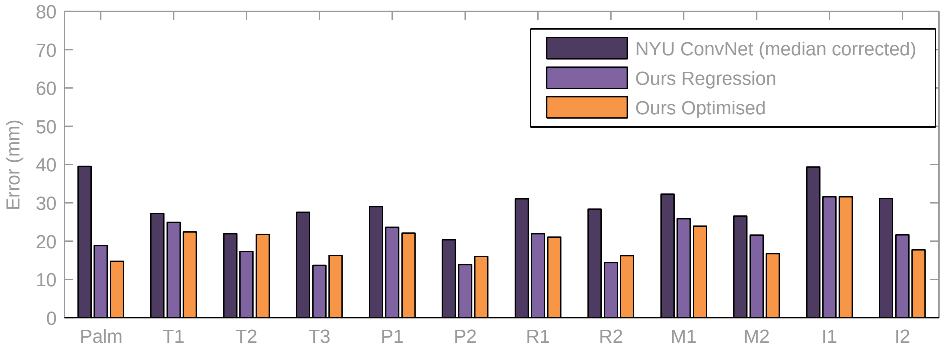


Figure 3.8: **Average error per joint.** Results on the NYU dataset (Actor 1) are compared to the hybrid *NYU ConvNet* (Tompson et al., 2014) approach.

3.4.3 Influence of the training set size

To investigate whether our contributions are effective in reducing the labeling effort, we evaluate how the results are affected when only training on a subset of the full training set. For this set of experiments we employ the NYU dataset and take a subsample of the original training set for training the joint regression model. More specifically, we investigate how the results are affected when changing the training set size, n. That is, we subsample different numbers of training samples and re-train the model only on the specific subset for each experiment.

Figure 3.9 shows how the mean joint error (ME) changes for different training set sizes. We see that our hybrid method improves the results over a number of different training set sizes. It especially improves the ME for a small number of training samples, while the ME for the full set even becomes slightly worse.

Looking closer at the differences between the ME and the success rate for the full training set, we find that these differences are very related to the differences of the methods and what they optimize for. More specifically, from the plot of the success rates in Figure 3.7 above, we see that the plain Regression Forest achieves very accurate estimates for a small number of frames, while the hybrid approach can improve the results for a large number of less accurately estimated frames.

When employing the full training set, despite the differences in the success rate (Figure 3.7), the mean joint error (ME) is similar for both methods. This indicates that the plain Regression Forest yields a high error for a small number of joints in many frames. At the same time many other joints are estimated accurately. Hence, the overall mean joint error is not significantly disturbed, but the frame-based success rate clearly shows the issue. This

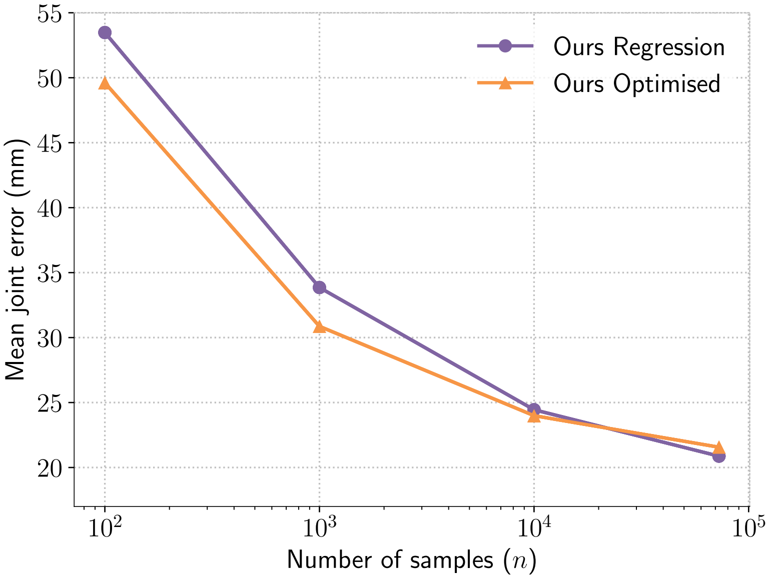


Figure 3.9: **Ablation experiments over n (mean joint error).** The mean joint error (ME) for the plain joint regression model and our hybrid method. For each n the results are averaged over ten experiments with different training subsamples.

becomes even clearer when comparing the frame-based success rate (FS) over different training set sizes in Figure 3.10. For this metric the improvement is much more pronounced and consistent over the number of training samples. The results point out that the large regression errors in many of the frames can be reduced by our integrative hybrid method considering the holistic joint configuration in each individual frame.

Since we take random subsamples from the full training set to obtain training sets of different sizes, the selection of the subsample affects the results. To investigate this effect we repeat each experiment ten times and study how our model-based optimization procedure affects the results of each experiment. In Figure 3.11 we plot the change of the ME, Δ_{ME}, and the change of the FS, Δ_{FS}, for each experiment after optimization. That is, Δ_{ME} is simply the difference between the ME for the Regression Forest, m_f, and the hybrid method, m_h: $\Delta_{ME} = m_h - m_f$ and accordingly $\Delta_{FS} = s_h - s_f$, where s_h and s_f are the FSs for the Regression Forest and the hybrid method, respectively. This means, improved results by the hybrid method are represented by a negative change of the ME, $\Delta_{ME} < 0$, and a positive change of the FS, $\Delta_{FS} > 0$. Comparing Figure 3.11a and 3.11b, we again see the discussed differences between the different metrics for the comparison of the data-driven and hybrid method. More interestingly, despite the significant variation for a small number of training samples, we see that our hybrid method consistently improves the results – independent of

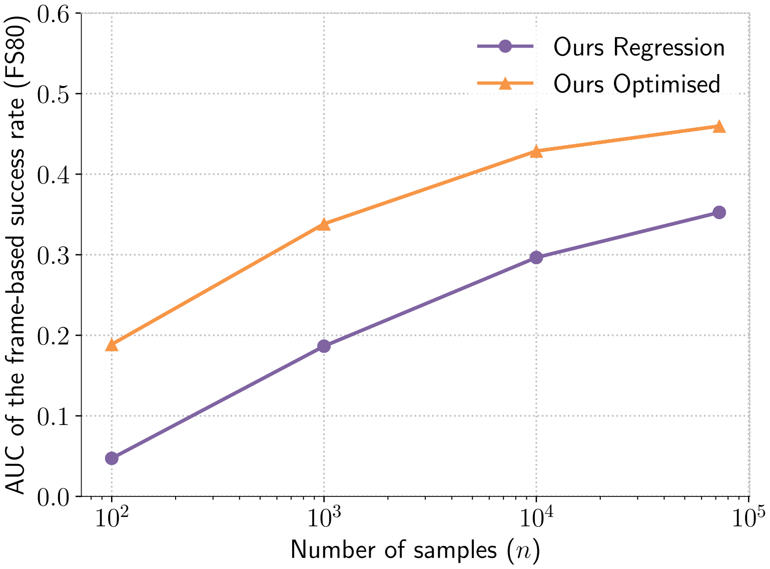


Figure 3.10: **Ablation experiments over n (success rate).** The area under the frame-based success rate curve up to a distance threshold of 80 mm (FS80) for the plain joint regression model and our hybrid method. For each n the results are averaged over ten experiments with different training subsamples.

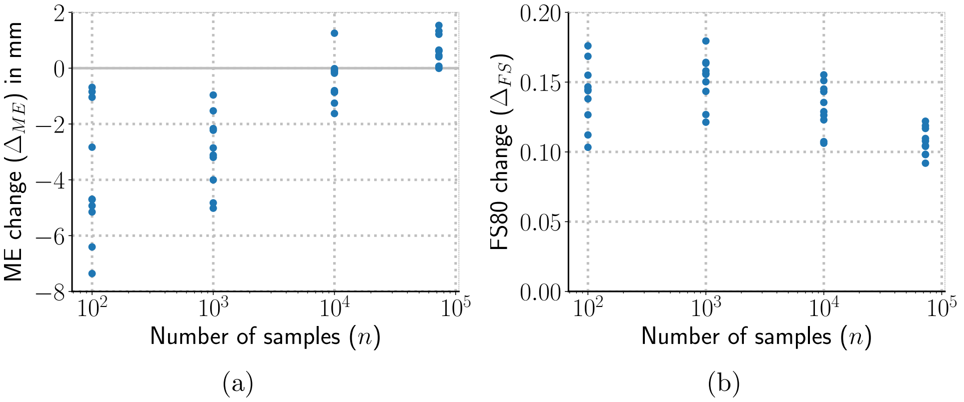


Figure 3.11: **Affect of optimization on individual results.** Changes of (a) the mean joint error (ME) and (b) the frame-based success rate (FS80) after optimization. We conducted ten experiments for each training set size n and each datapoint shows the change for an individual experiment. Note, improved results are represented by a negative change of the ME and a positive change of the FS80.

the selection of the training set. Furthermore, we can see that the potential performance gain for our hybrid method decreases with increasing training set size, *i.e.*, the potential gain is larger for smaller training set sizes.

The improvement obtained by our hybrid method is also affected by the optimization budget, which we set based on a trade-off between accuracy and runtime. The runtime, however, is also strongly affected by the actual

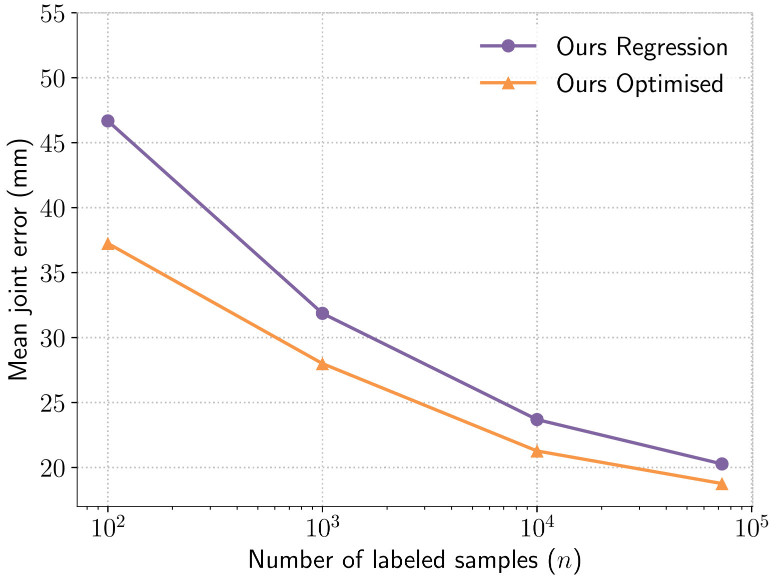


Figure 3.12: **Ablation experiments for the best runs over different training set sizes.** The best mean joint error (ME) for the plain joint regression model compared to the result after optimization with a high budget.

implementation and hardware resources. Hence, for a final experiment we aim to investigate the potential improvement when ignoring runtime considerations. Together with this we investigate the potential improvement when already starting from a strong baseline, *i.e.*, for the baseline we assume that we are able to select a “good“ training set for the data-driven approach. That is, we selected the best performing out of ten Regression Forests for each n and employ a high optimization budget for the model-based optimization part. The results are shown in Figure 3.12. As expected, the improvement is generally larger than in the case where we compared the average results over multiple experiments. However, this is despite the fact that we already start from a higher baseline. Moreover, we see that the improvement is consistent over all n. That is, the model-based optimization procedure can consistently improve even the best obtained results from the Regression Forest in terms of the mean joint error (ME).

3.4.4 Qualitative results

Finally, we show some qualitative results and error cases of our method. Figure 3.13 and 3.14 show input depth maps and the estimated poses rendered from different view points from the ICVL and the NYU dataset, respectively.

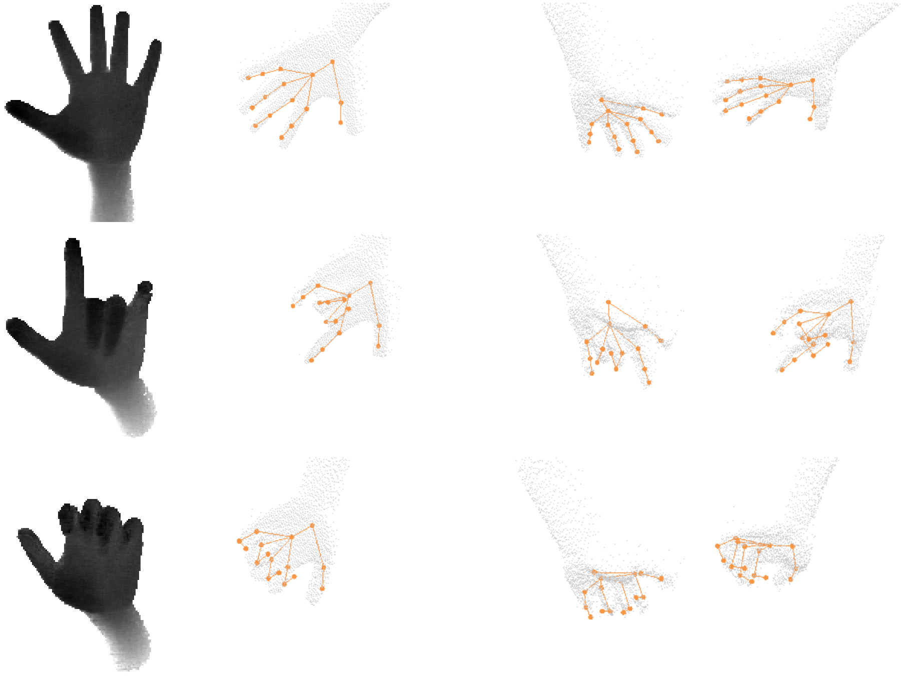


Figure 3.13: **Qualitative results for the ICVL dataset.** Each row shows the input and result for a single frame. The left most column shows the input depth maps, the remaining columns show the estimated pose rendered from different view points.

Looking at the error cases we can see that often the joints of a single finger are estimated incorrectly, while the overall structure of the hand was estimated rather well. Figure 3.15 shows two examples for error cases. High errors for single joints are often induced if the parameters of a joint earlier in the kinematic chain are incorrectly estimated. The inaccurate estimate for a joint obviously affects all subsequent joints in the chain.

Such error cases can arise when the model-based optimization procedure ends up in a bad local minima or didn't properly converge due to the stochasticity, respectively. Possible solutions to such error cases could, for example, be provided by a better initialization, switching to a gradient based optimization towards the end of the optimization procedure, or a higher optimization budget, which could be enabled by a more efficient, parallel implementation.

3.5 Conclusion

In this chapter, we introduced a hybrid approach for 3D hand pose estimation based on a single depth frame. A Regression Forest delivers sev-

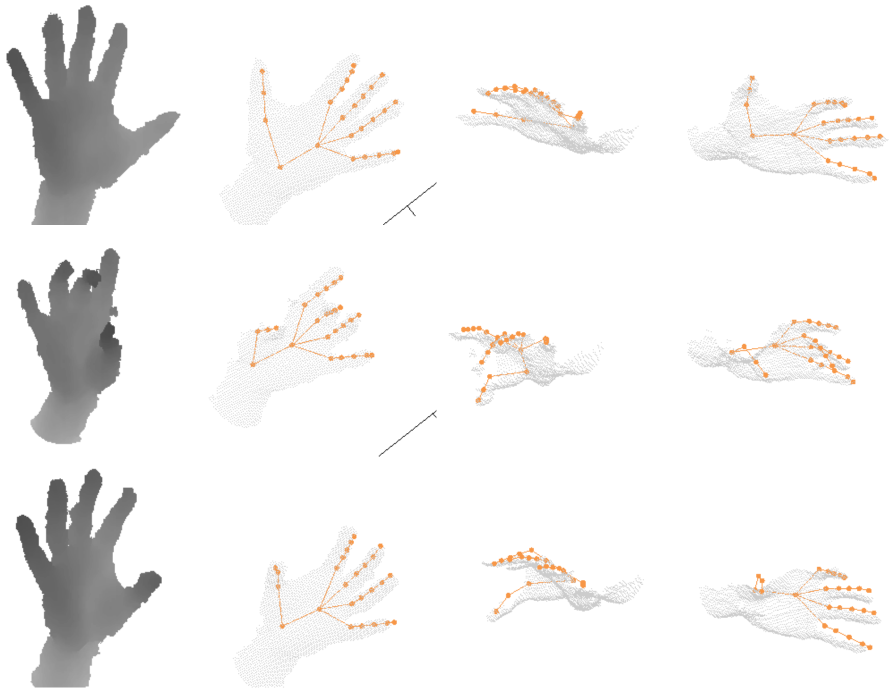
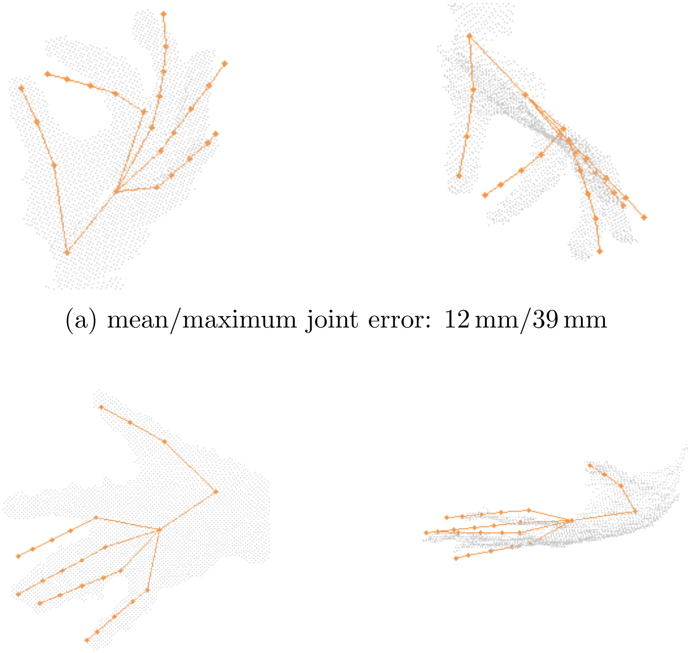


Figure 3.14: **Qualitative results from the NYU dataset.** Each row shows the input and result for a single frame. The left most column shows the input depth maps, the remaining columns show the estimated pose rendered from different view points. (Note, for this dataset, for the point cloud renderings one axis was swapped with respect to the depth image)

eral proposals for each hand joint position. Then, model-based optimization is responsible for estimating the best fit of a 3D hand model to the joint proposals obtained through regression. Thus, optimization exploits the inherent uncertainty of the data-driven regression. As a result, the introduced method delivers anatomically valid solutions (which most purely data-driven methods fail to provide) without unrecoverable track losses or a need for proper initialization (as happens with most purely model-based approaches). At the time of first publication (Poier et al., 2015), the introduced method has been shown to achieve state-of-the-art performance. Moreover, we showed that by employing such a hybrid approach similar accuracy can be achieved using a reduced number of training samples. This is proven by quantitative experiments on several datasets and in comparison to the respective baselines as well as representative methods from all categories (model-based, data-driven, hybrid).

While the introduced method provides the possibility to reduce the labeling effort, a significant amount of labeled training samples is still necessary to achieve meaningful results. This is due to the fact that the model-based



(a) mean/maximum joint error: 12 mm/39 mm

(b) mean/maximum joint error: 14 mm/58 mm

Figure 3.15: **Error cases for the NYU dataset.** Each row shows the results for a single frame rendered from two different view points.

optimization scheme cannot fix gross errors, but is limited to the "initialization" provided by the data-driven part. And the data-driven part is unable to provide an estimate close to the true solution for poses too far from the training set distribution. This situation clearly hampers performance.

To overcome the issues, in the next chapter we focus on improving the data-driven part for the case when only a small number of labeled samples is available. In order to make the data-driven part less restricted to the distribution of labeled training data, we introduce a way to exploit unlabeled data. We will show that unlabeled data – if properly exploited – provide a way to make the model learn about the structure of poses for which no labeled samples are in the training set.

CHAPTER 4

Exploiting unlabeled data

Contents

For the task of hand pose estimation, the best performing methods have recently relied heavily on models learned from data (Guo et al., 2017; Oberweger et al., 2015b; Sun et al., 2015; Supancic et al., 2015). Even methods which employ a manually created hand model to search for a good fit with the observation, often employ such a data-driven part as initialization or for error correction (Krejov et al., 2017; Tang et al., 2015; Taylor et al., 2016) – similar to the approach we introduced in the previous chapter. Unfortunately, data-driven models require a large amount of labeled data, covering a sufficient part of the pose space, to work well.

However, for the task of estimating the pose of articulated objects, like the human hand, it is especially expensive to provide accurate annotations for a sufficient amount of real world data. The articulated structure and specific natural movements of the hand frequently cause strong self-occlusions. Together with the many 3D points to be annotated, this makes the annotation procedure a huge effort for human annotators.

A largely unexplored direction to cope with this challenge is to exploit unlabeled data, which is easy to obtain in large quantities. We introduce

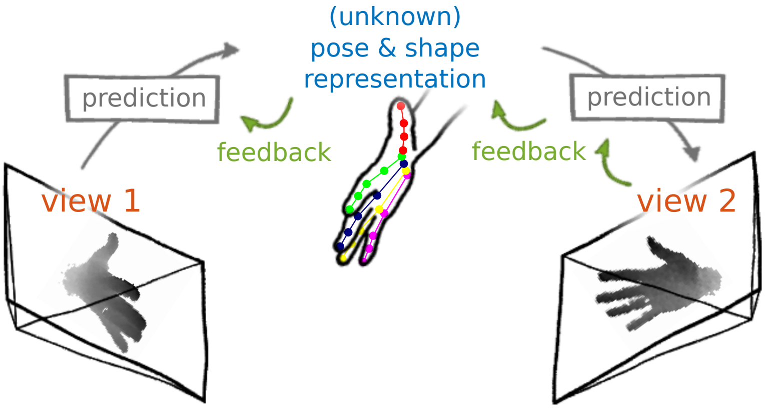


Figure 4.1: **Sketch for learning a pose specific representation from unlabeled data.** We learn to predict a low-dimensional latent representation and, subsequently, a different view of the input, *solely* from the latent representation. The error of the view prediction is used as feedback, enforcing the latent representation to capture pose specific information without requiring labeled data.

a step towards closing this gap with a method that can exploit unlabeled data by making use of a specific property of the pose estimation task. We rely on the observation that pose parameters[1] are predictive for the object appearance of a known object from any viewpoint. That is, given the pose parameters of a hand, the hand's appearance from any viewpoint can be estimated. The observation might not seem helpful upfront, since it assumes the pose – which we want to estimate – to be given. However, the observation becomes helpful if we capture the scene simultaneously from different viewpoints.

By employing a different camera view, we can guide the training of the pose estimation model (see Figure 4.1). The guidance relies on the fact that from any set of pose parameters, which accurately specify the pose and rough shape of the hand, we necessarily need to be able to predict the hand's appearance in any other view. Hence, by capturing another view, this additional view can be used as a target for training a model, which itself guides the training of the underlying pose representation.

More specifically, the idea is to train a model which – given the first camera view – estimates a small number of latent parameters, and subsequently predicts a different view solely from these few parameters. The intuition is that the small number of parameters resemble a parameterization of the pose. By learning to predict a different view from the latent parameters, the latent parameters are enforced to capture pose specific information. Framing the problem in this way, a pose representation can be learned just by

[1] For the sake of clarity, here, *pose parameters* denote the parameters defining the skeleton, including its size, as well as a rough shape

capturing the hand simultaneously from different viewpoints and learning to predict one view given the other.

Given the learned low-dimensional pose representation, a rather simple mapping to a specific target (*e.g.*, joint positions) can be learned from a much smaller number of training samples than required to learn the full mapping from input to target. Moreover, when training jointly with labeled and unlabeled data, the whole process can be learned end-to-end in a semi-supervised fashion, achieving similar performance with one order of magnitude less labeled samples. Thereby, the joint training regularizes the model to ensure that the learned pose representation can be mapped to the target pose space using the specified mapping.

In this chapter, which is largely based on our previous publication on *Learning Pose Specific Representations by Predicting Different Views* (Poier et al., 2018), we show the specificity of the learned representation and its predictiveness for the pose in qualitative and quantitative experiments. Trained in a semi-supervised manner, the introduced method consistently outperforms its fully supervised counterpart, as well as the state-of-the-art in hand pose estimation – even if all available samples are labeled. For the more practical case, where the number of unlabeled samples is larger than the number of labeled samples, we find that the introduced method performs on par with the baseline, even with one order of magnitude less labeled samples.

4.1 Related work

As discussed above, traditionally, works on hand pose estimation have been divided into model-based and data-driven approaches. Model-based approaches (de La Gorce et al., 2011; Melax et al., 2013; Oikonomidis et al., 2011a; Roditakis et al., 2017; Wu et al., 2001) search to parameterize a manually created hand model in each frame such that it best fits the observation. These approaches usually need to rely on an initialization, *e.g.*, from previous frames, and thus, have problems to recover if pose estimation fails once. Data-driven approaches, on the other hand, learn a mapping from the input frame to a target pose from a usually large number of annotated training samples (Guo et al., 2017; Keskin et al., 2012; Oberweger and Lepetit, 2017; Tang et al., 2014). These approaches assume that the poses seen at test time are at least roughly covered by the training set and will otherwise fail to deliver a good estimate. With the desire to combine the merits of both strands, hybrid approaches, combining both strands, have been developed (Mueller et al., 2017; Taylor et al., 2016; Zhou et al., 2016). But, as pointed out in the previous chapter, the effectiveness of hybrid approaches is again crucially affected by the density of the annotations available for training the data-driven part.

Data annotation To provide a large number of labeled samples, semi-automatic or sometimes even automatic methods were employed to construct the relevant publicly available training sets. Most often model-based approaches with the above mentioned issues were used to provide (initial) annotations, which were manually corrected (Sun et al., 2015; Tang et al., 2014; Tompson et al., 2014). Other efforts include the development of an annotation procedure (Oberweger et al., 2016) to propagate annotations to similar frames, or attaching 6D magnetic sensors to the hand (Wetzler et al., 2015; Yuan et al., 2017), which resulted in the largest dataset to date (Yuan et al., 2017). These efforts underline the difficulties to provide sufficient labeled data, hampering novel applications, which might rely on different viewpoints or sensors.

Learning from unlabeled data At the same time, capturing unlabeled data is easy, and considering the way how we make use of such unlabeled data, several strands of prior work are related to our method. The scheme of predicting another view from the learned latent representation is, *e.g.*, akin to the concept of autoencoders, where the input is reconstructed from the latent representation (Hinton and Salakhutdinov, 2006; Vincent et al., 2008). Instead of reconstructing the input, we learn to predict a different view. This enables the model to capture pose specific representations as the results in Sec. 4.4.3 clearly point out.

Similarly, the method presented in this chapter is also related to a strand of works on representation learning from unlabeled data which obtain the training target by adapting the input. Such works, *e.g.*, transform the input and have the model learn to predict the transformed version given the original input (Hinton et al., 2011), or split the input data into parts and have the model learn relations between the parts (Doersch et al., 2015; Owens et al., 2016; Pathak et al., 2016; Zamir et al., 2016; Zhang et al., 2017). For instance, Doersch et al. (2015) learn to predict the relative position of patches sampled from an image, which should be possible if a model has learned to extract semantics. Similarly, this has been targeted by, *e.g.*, relating tracked patches (Wang and Gupta, 2015), solving jigsaw puzzles (Noroozi and Favaro, 2016) or colorizing images (Larsson et al., 2016). While our method can be considered similar in spirit, our main objective is to learn a pose specific representation in the latent space, for which a crucial enabler is to employ multiple viewpoints.

Learning from multiple views Researchers have also employed multiple camera views to learn depth prediction or 3D object reconstruction from unlabeled data (Garg et al., 2016; Jayaraman et al., 2017; Xie et al., 2016). Garg et al. (2016) propose an approach to monocular depth estimation, for which the loss is based on the photo consistency of the projected pixels in the second view of a stereo image pair. Similarly, Xie et al. (2016) target generating a stereo pair from a single view. Several works add upon this line of research, *e.g.*, by incorporating sparse and noisy depth labels (Kuznietsov et al., 2017), adding a left-right consistency constraint (Godard et al., 2017), jointly estimating camera pose and depth (Zhou et al., 2017a), or learning to reconstruct full 3D (Tatarchenko et al., 2016; Tulsiani et al., 2017; Yang et al., 2015).

In these works the desired target (*e.g.*, depth or disparity) can directly be linked to the training loss via geometric relations and, therefore, only the intermediate latent representations have to encode some kind of semantics of the scene and objects therein. In our case, the target itself is more explicit semantic (*e.g.*, joint positions or labels, resp.) and we show how to formulate the task such that our learned latent representation closely resembles what we are targeting, namely the pose. The formulation also clearly differentiates our method from generic multi-view learning approaches, which we discussed in Chapter 2, like Canonical Correlation Analysis (CCA) and its variants.

Semi-supervised learning for hand pose estimation Little work has exploited unlabeled samples for hand pose estimation. To the best of our knowledge, there are only some notable exceptions (Neverova et al., 2015; Tang et al., 2013; Wan et al., 2017): Tang et al. (2013) built a discriminative approach which relies on a large synthetic training set and correspondences between synthetic and real samples. Similarly, Neverova et al. (2015) establish correspondences via an intermediate representation of part segmentations. For their approach, they do not need pixelwise labels for real samples, but still require joint annotations. On the contrary, Wan et al. (2017) incorporate entirely unlabeled data by drawing from advances in generative modeling within a semi-supervised approach. While elegant and well set up, neither of these approaches exploit the observation that the pose is predictive for the appearance from any known view.

View synthesis for hand pose estimation Another notable work on hand pose estimation, we draw inspiration from, is the work from Oberweger et al. (2015b). They aim to reconstruct the input view of the hand from previously estimated joint positions, and subsequently learn to generate an update for the pose estimate based on the discrepancy between the input and the reconstruction (akin to supervised descent methods (Sheerman-Chase et al., 2013; Xiong and De la Torre, 2013)).

In contrast to the method introduced in this chapter, however, they aim to reconstruct the same view directly from previous estimates of the joint positions (without capturing shape information). Consequently, their approach is fully supervised, *i.e.*, it requires joint annotations for each sample. In our method, we do not require pose annotations, but exploit the information we get from an additional view point, which is crucial for the training process, as we will show in our experiments. Nevertheless, inference is straight forward with our method, *i.e.*, we neither require an iterative procedure and generation of images as in (Oberweger et al., 2015b), nor need a second view at test time.

4.2 Formulating the observations

Our idea is based on the observation that a hand pose representation, $\boldsymbol{\theta}$, which includes parameters for the hand's size and shape, is predictive for the hand's appearance, $\mathbf{x}^{(i)}$, from any known view i. Let $\mathcal{T} \subset \mathbb{R}^{d_\mathcal{T}}$ denote the set of possible poses or pose representations of dimensionality $d_\mathcal{T}$, *i.e.*, $\boldsymbol{\theta} \in \mathcal{T}$, and similarly, $\mathcal{X} \subset \mathbb{R}^{d_\mathcal{X}}$ be the set of possible input images of dimensionality $d_\mathcal{X}$, *i.e.*, $\mathbf{x}^{(i)} \in \mathcal{X}$. Then – based on our observation – we assume that there exists a view specific mapping, $g_i^*\colon \mathbb{R}^{d_\mathcal{T}} \to \mathbb{R}^{d_\mathcal{X}}$, such that

$$\mathbf{x}^{(i)} = g_i^*(\boldsymbol{\theta}), \qquad \forall \boldsymbol{\theta} \in \mathcal{T}. \tag{4.1}$$

Nevertheless, for our task we do not know the pose. The pose is what we are searching for. Given an image of a hand $\mathbf{x}^{(i)}$ we want to find the pose of the hand. That is, we search for a mapping $f_i^*\colon \mathbb{R}^{d_\mathcal{X}} \to \mathbb{R}^{d_\mathcal{T}}$ from the input image to the pose[2]:

$$\boldsymbol{\theta} = f_i^*(\mathbf{x}^{(i)}), \qquad \forall \mathbf{x} \in \mathcal{X}. \tag{4.2}$$

Clearly, given the two mappings, f^* and g^*, by subsequently applying them we can map from input to pose and back. But we can also see that we can directly map from one view to another. That is, given an input image of the hand, $\mathbf{x}^{(i)}$, from view i, we can use the mappings to compute the

[2] To avoid cluttering the notation, we ignore that such a mapping is not always unique, given only a single view. In theory, we could formulate $\boldsymbol{\theta}$ as a random variable, describing a distribution, we could sample from.

hand's appearance $\mathbf{x}^{(j)}$, from any known view j:

$$\mathbf{x}^{(j)} = g_j^*\big(f_i^*(\mathbf{x}^{(i)})\big). \tag{4.3}$$

In our case, the mappings f^* and g^* are unknown. We can, however, capture the scene simultaneously from two different views i and j. Given the data from two views, $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, we can formulate our problem as finding a mapping from one view to another and learn both mappings by solving this task. While, for our ultimate goal we only need the mapping f^* from the input to the pose, the second mapping is required to effectively exploit the supervision provided by a different view. Hence, we use the task of learning a mapping from one view to the other as a surrogate task for learning a mapping to a latent representation, which resembles the pose.

Note, for $i = j$, Eqn. (4.3) essentially specifies an autoencoder. From our empirical investigation (see Sec. 4.4.3) we find that in this case the model is unable to learn a representation specific to the pose. However, for $i \neq j$ and a sufficient amount of (unlabeled) data we find that it is easy to constrain the model such that the latent representation captures pose information. Hence, the crucial case, which we are investigating in this chapter, is the case $i \neq j$.

4.3 Implementing the observations

To implement our observations we want to learn the two mappings, f^* and g^*, from data. We do so by employing a Convolutional Neural Network (CNN) with an encoder-decoder architecture. To formalize our method, we denote the learned estimates of the "true" mappings f^* and g^*, f and g, respectively. The encoder f_i receives input $\mathbf{x}^{(i)}$ from view i and its output represents the desired latent representation $\boldsymbol{\theta}$. The latent representation is at the same time the input for the decoder g_j, which produces the view $\mathbf{x}^{(j)}$ given $\boldsymbol{\theta}$. Without loss of generality we assume the captured images, $\mathbf{x}$, to be depth images. Note that, while, for color-only input the appearance is affected by additional factors like skin color or illumination, the basic observations still hold.

In the basic model, we train our system to predict a different view $\mathbf{x}^{(j)}$, which we capture for training. The training loss, ℓ_{u}, for this model can thus be formulated as a reconstruction loss

$$\ell_{\mathrm{u}} = \ell_{\mathrm{recon}}\big(\hat{\mathbf{x}}^{(j)}, \mathbf{x}^{(j)}\big), \tag{4.4}$$

where $\hat{\mathbf{x}}^{(j)}$ is the model's prediction for view j, given input $\mathbf{x}^{(i)}$ from view i, *i.e.*,

$$\hat{\mathbf{x}}^{(j)} = g_j\big(f_i(\mathbf{x}^{(i)})\big). \tag{4.5}$$

For the reconstruction loss ℓ_{recon} we experimented with the L^1-, L^2- and Huber-norm and found the L^1-norm to yield the best results.

Ideally, we want the latent representation, $\boldsymbol{\theta} = f_i(\mathbf{x}^{(i)})$, to be very specific for the pose, not capturing any unnecessary information. The loss itself does not constrain the latent representation to fulfill such a requirement. We can, however, constrain the latent representation in a very simple – though effective – way: We assume that the smallest possible representation which is predictive for the appearance of any known view, other than the input, will, crucially, contain a representation resembling the pose.

A low-dimensional representation of the pose is often given by the joint positions. However, since there are many dependencies between the joints, the pose can even be represented by a lower-dimensional subspace. While works on hand modeling (Albrecht et al., 2003; Lin et al., 2000) give an indication for the size of such a low-dimensional subspace, we investigate the size best matching our requirements in the experimental section (Sec. 4.4.3). The representation should contain only little additional information which could obfuscate the pose representation and, thus, hamper learning a mapping to any target pose representation as discussed in the next section.

4.3.1 Learning from labeled and unlabeled data

To map from the latent representation space to the desired target space (*e.g.*, joint positions) we add a single linear layer to our encoder-decoder architecture. Using a linear layer is a very common way to evaluate learned representations (Coates et al., 2011; Dosovitskiy et al., 2014; Noroozi et al., 2017). In this way only a limited amount of additional parameters need to be learned from labeled data. We enforce the latent representation to suffice this linear map by training the encoder, which maps from input to the latent representation, jointly with labeled and unlabeled data in a semi-supervised manner. That is, labeled samples guide the training of the latent representation such that it suffices the linear mapping.

The architecture for semi-supervised training is depicted in Figure 4.2. The parameters of the linear layer from the latent pose representation to the joint positions are only trained using labeled samples. All other parameters are trained using both labeled and unlabeled samples.

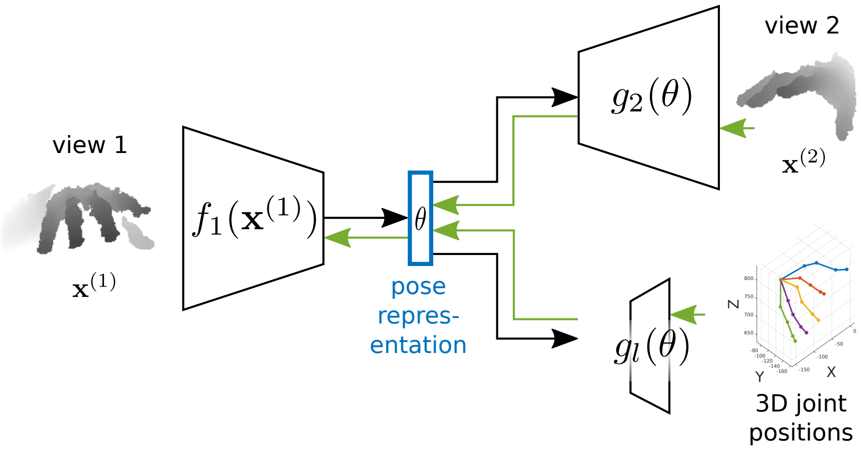


Figure 4.2: **Architecture sketch for semi-supervised learning.** The input view, $\mathbf{x}^{(1)}$, is mapped to the latent representation, $\boldsymbol{\theta}$, by the encoder f_1. Solely based on $\boldsymbol{\theta}$, the decoder g_2 is required to generate a different view 2 of the input. At the same time the latent representation is ensured to suffice a linear mapping, g_l, to the 3D joint positions by employing labeled samples. This is illustrated by the green paths depicting the gradient flow to the latent representation and, consequently, to the encoder.

The semi-supervised loss function, ℓ_{semi}, is a combination of the loss from unlabeled and labeled data:

$$\ell_{\text{semi}} = \ell_{\text{u}} + \lambda_{\text{l}}\, \ell_{\text{l}}, \tag{4.6}$$

where λ_{l} is a weighting factor, which is set to zero for unlabeled samples. Due to the difficulties with labeling hand poses in 3D, essentially all current datasets exhibit at least some label errors. For robustness to such errors, we employ the sum of the Huber loss (Huber, 1964) for individual joint errors. Note, this is different from the standard use of the Huber loss. That is, in our case

$$\ell_{\text{l}} = \sum_m \ell_{\text{Huber}}\left(\|\mathbf{y}_m - \hat{\mathbf{y}}_m\|_2\right), \tag{4.7}$$

where $\hat{\mathbf{y}}_m$ denotes the estimated position of the m-th joint, $\mathbf{y}_m$ the corresponding ground truth position, $\|\cdot\|_2$ the L^2-norm of the argument and

$$\ell_{\text{Huber}}(d) = \begin{cases} 0.5\, d^2 & \text{if } d < \epsilon \\ \epsilon\,(d - 0.5\,\epsilon) & \text{otherwise.} \end{cases} \tag{4.8}$$

Note that d, the input to the Huber loss, is always positive in our case.

4.3.2 Beyond pixelwise auxiliary objectives

In the previous section the objective for the decoder is based on a simple reconstruction loss. In this way, the decoder is penalized for any deviation from the second view's exact pixel values. However, more important for our task is the global structure of the image as this is affected crucially by the pose. That is, the decoder spends representational power on estimating exact pixel values, which are of little interest to us.

Instead of predicting exact pixel values, the goal of our method is to capture the latent representation determining the pose of the hand shown in the image. The adversarial training procedure introduced by Goodfellow et al. (2014) points out a way to provide loss functions beyond pixelwise losses. This also appears interesting for our goal since such approaches have been shown to learn interpretable latent representations (see, *e.g.*, (Radford et al., 2016)). The training procedure corresponds to a minimax two-player game, where each player is implemented by a neural network. A generator network aims to generate samples from the data distribution and a discriminator network aims to discriminate generated samples from real samples. In this game, the loss for the generator is essentially provided by the discriminator, thus overcoming the need for explicit supervision, *e.g.*, from corresponding target images.

Using this idea, we can train the decoder of our method to match the distribution of real images, but lessen the focus on raw pixel differences. We do so by adding an additional adversarial term to the loss in Eq. (4.6),

$$\ell_{\mathrm{semi}} = \ell_{\mathrm{u}} + \lambda_{\mathrm{l}}\, \ell_{\mathrm{l}} + \lambda_{\mathrm{a}}\, \ell_{\mathrm{a}}, \tag{4.9}$$

where λ_{a} is a weighting factor and ℓ_{a} can be intuitively interpreted as how "unreal" the discriminator network h thinks a generated sample $\hat{\mathbf{x}}$ is. That is, since this yielded the best results, we define ℓ_{a} inspired by Least Squares GAN (Mao et al., 2017) as

$$\ell_{\mathrm{a}} = \frac{1}{2}\left(h_j(\hat{\mathbf{x}}^{(j)}) - l_r\right)^2, \tag{4.10}$$

where $l_r \in \mathbb{R}$ is the label value for real samples. The objective for the discriminator, on the other hand, is to push the real samples towards l_r and generated samples towards a different label value $l_g \in \mathbb{R}$, *i.e.*,

$$\ell_{\mathrm{h}} = \frac{1}{2}\left(h_j(\mathbf{x}^{(j)}) - l_r\right)^2 + \frac{1}{2}\left(h_j(\hat{\mathbf{x}}^{(j)}) - l_g\right)^2. \tag{4.11}$$

In our case we set $l_r = 1$ and $l_g = 0$.

The loss in Eq. (4.9) requires the decoder g to output an image closely resembling the image of the second view since the reconstruction loss is still part of its objective. Nevertheless, g is enforced to focus more on the overall structure of the image through the loss term provided by the discriminator.

For adversarial training it has been shown that the discriminator can be improved – and thus provide better feedback – by conditioning it on additional input (Mirza and Osindero, 2014). For our task we can condition the discriminator on the input from the first view and/or, in case of semi-supervised training, the pose. That is, the input for the discriminator is not only the generated or captured image for the second view j (*i.e.*, $\hat{\mathbf{x}}^{(j)}$ or $\mathbf{x}^{(j)}$, resp.) but also the input to the generator (*i.e.*, $\mathbf{x}^{(i)}$) and/or the pose $\mathbf{y}$. For the case of conditioning on the pose, we provide the estimated pose $\hat{\mathbf{y}}$ for generated samples and the ground truth pose $\mathbf{y}$ for real samples. In our experiments below we evaluate all described conditioning types.

4.3.3 Implementation details

Similar to other works (Krejov et al., 2017; Tang et al., 2014) we assume the hand to be the closest object to the camera, and compute its center of mass (CoM), which is also provided as additional input to the decoder, g. We then crop a region with equal side length in each direction around the CoM, resize it to 64 × 64 pixels and normalize the depth values within a fixed range to be between -1 and 1. These crops form the input to our method.

Our method does not rely on a specific choice of the network architecture. For our experiments, we implemented our encoder and decoder networks based on the architecture developed for DCGAN (Radford et al., 2016), since it is a well developed architecture, which is comparably "lightweight" and designed for image synthesis. We base our encoder f and our discriminator h on the discriminator and our decoder g on the generator of the original publicly available implementation[3]. We only interchange the positions of the ReLUs (Fukushima, 1980; Nair and Hinton, 2010) and leaky ReLUs (Maas et al., 2013) since we want to ease gradient flow through the decoder, put a hyperbolic tangent (tanh) activation function at the end of the decoder to ensure that the output can range between -1 and 1, and adapt the input and output dimensions accordingly.

We train our model with *Adam* (Kingma and Ba, 2015) for 100 epochs using a batch size of 128 and a learning rate of 10^{-4}. For semi-supervised

[3] `https://github.com/soumith/dcgan.torch`

learning we obtained the best results with $\lambda_l = 10$. Our *PyTorch*[4] implementation is publicly available[5].

4.4 Experiments

To prove the applicability of the proposed method we perform qualitative and quantitative experiments on different datasets. We investigate the representations learned from unlabeled data (*c.f.* Sec. 4.3) in Sec. 4.4.3. Subsequently, we present the results for semi-supervised learning (*c.f.* Sec. 4.3.1), compare to the state-of-the-art in hand pose estimation, and provide evidence for the effectiveness of our training procedure in an ablation study (Sec. 4.4.4).

4.4.1 Datasets

We evaluate on two different datasets. Firstly, we test on the NYU hand pose dataset (Tompson et al., 2014), which, to the best of our knowledge, is the only public dataset providing multiple views for the training and test set. For a broader empirical analysis of our method we additionally provide a novel multi-view dataset[6].

NYU hand pose dataset The NYU dataset provides a training set with 72,757 frames from a single actor and a test set with 8,252 frames from two actors. It was captured with structured light based RGBD cameras. The additional cameras captured the scene from side views. Originally, the additional cameras were employed to mitigate issues with self-occlusions during annotation; for our evaluation the additional camera views enable us to compare our method on a standard dataset. Unfortunately, the side view camera locations were changed several times during training set acquisition and no camera pose information is provided. Therefore, we searched for a part of the training set with approximately similar camera setup and found 43,641 frames (∼60% of the original training set), which we used as a training set for our experiments. For validation and testing, we use the full sets from the original dataset. We denote the reduced training set with consistent setup by *NYU-CS*.

Multi-view hand pose dataset We captured the dataset for typical user interaction scenarios in front of a large screen with a Time-of-Flight (ToF)

[4] `http://pytorch.org`

[5] Project webpage with code, data and additional material can be found at `https://poier.github.io/PreView`

[6] See footnote 5

camera mounted at each of the two top corners of the screen. We captured the two cameras synchronously and captured poses needed for typical gestures like swiping, pointing or waving. While the set of poses is restricted, we aimed to capture each pose in all meaningful hand orientations and ended up with 63,701 frames from 14 different actors. Since the goal of our novel dataset is to investigate semi-supervised learning where only a small fraction of the available samples is labeled, we only labeled a representative subset from a few actors. To this we employed the method in (Oberweger et al., 2016), which tries to find a subset of frames covering the pose space well. Overall 526 frames from 7 out of the 14 actors were manually annotated. We split the labeled data in 289 frames for training and validation (189/100) and 237 for testing. We denote the resulting multi-view hand pose dataset *MV-hands*.

4.4.2 Metrics

For the evaluation, we employ three commonly used metrics: the mean joint error (ME) as well as the joint- and frame-based success rate (JS/FS). The ME denotes the average distance between the estimated and ground truth joint positions in millimeter (mm). The JS is the fraction of joints which were estimated within a certain distance to the ground truth joint position. The FS is stricter and gives the fraction of frames for which all joints have been estimated within a certain distance to the ground truth position (Taylor et al., 2012).

For hand pose estimation, researchers often employ curves of the success rates over different distance thresholds. To express these curves with a single number, we compute the area under the curve (AUC) up to a specified threshold. We denote the AUC of the JS and FS up to a distance threshold of 80 mm by *JS80*, and *FS80*, respectively.

4.4.3 Representations learned from unlabeled data

In the following, we perform several experiments to investigate the effectiveness of representations learned from unlabeled data.

Linear mapping to joint positions

To quantitatively analyze the predictability of the pose given the learned latent representations, we follow the standard procedure for testing representations learned in an un-/self-supervised manner (Coates et al., 2011; Dosovitskiy et al., 2014; Noroozi et al., 2017; Zhang et al., 2016): We train the network using the respective pre-training method, *i.e.*, without pose

Table 4.1: **Results for representations learned from unlabeled data.** Mean joint error and standard deviation on the NYU-CS dataset for different pre-training methods and numbers of labeled samples, n.

n	Autoencoder	**PreView (Ours)**	
100	48.0 ± 0.76	$\mathbf{33.4} \pm 1.18$	-30.4%
1,000	47.2 ± 0.29	$\mathbf{29.6} \pm 0.32$	-37.3%
10,000	47.3 ± 0.08	$\mathbf{29.0} \pm 0.14$	-38.7%
43,640	47.1 ± 0.08	$\mathbf{29.0} \pm 0.09$	-38.4%

annotations, freeze all layers up to the latent pose representation and train a linear mapping from the latent representation to the target joint positions using annotated samples.

The results on the NYU-CS dataset are shown in Table 4.1. We compare our method to pre-training using an autoencoder because of its close relation. In particular, the autoencoder's target is the input view, whereas our method aims to predict a different view. For a fair comparison, we use the same architecture, *i.e.*, the same number of parameters and training algorithm for the autoencoder and the proposed method for predicting different views (*PreView*).

Here, we also investigate how the respective methods behave when the number of labeled samples, n, is smaller than the number of unlabeled samples, *i.e.*, only a subset of labeled samples is provided. In this case, we use a random subset of the data, which is the same for each method. For the case where the training set is small, the size of the validation set will – for a realistic scenario – be similarly small. To account for this, we also subsample the validation data. We fix the size of the validation set, $|\mathcal{V}|$, as a fraction of the size of the sub-sampled training set, $|\mathcal{L}|$. That is, we sample at most $|\mathcal{V}| \leq 0.3\,|\mathcal{L}|$ samples from the original validation set. We repeat this experiment 10 times with different random samples to investigate the effect of the sampling and report the average and standard deviation of the results in Table 4.1.

The results show that pre-training for view prediction yields a latent representation which is significantly more predictive for the pose than pre-training using an autoencoder. The improvement is consistent – independent of the ratio between labeled and unlabeled samples – and ranges between 30 and 40 percent.

On the other hand, qualitative inspection shows that the autoencoder yields cleaner reconstructions of the inputs, compared to the predictions of the second view of our method. In Figure 4.3 we compare output images of input reconstruction (autoencoder) and view prediction (PreView). Figure 4.4 shows more view prediction examples on the MV-hands dataset. We can observe that the reconstructions of the input are cleaner (*e.g.* for the fingers) than the predictions for different views. Obviously, reconstructing the input is an easier task than predicting a different view.

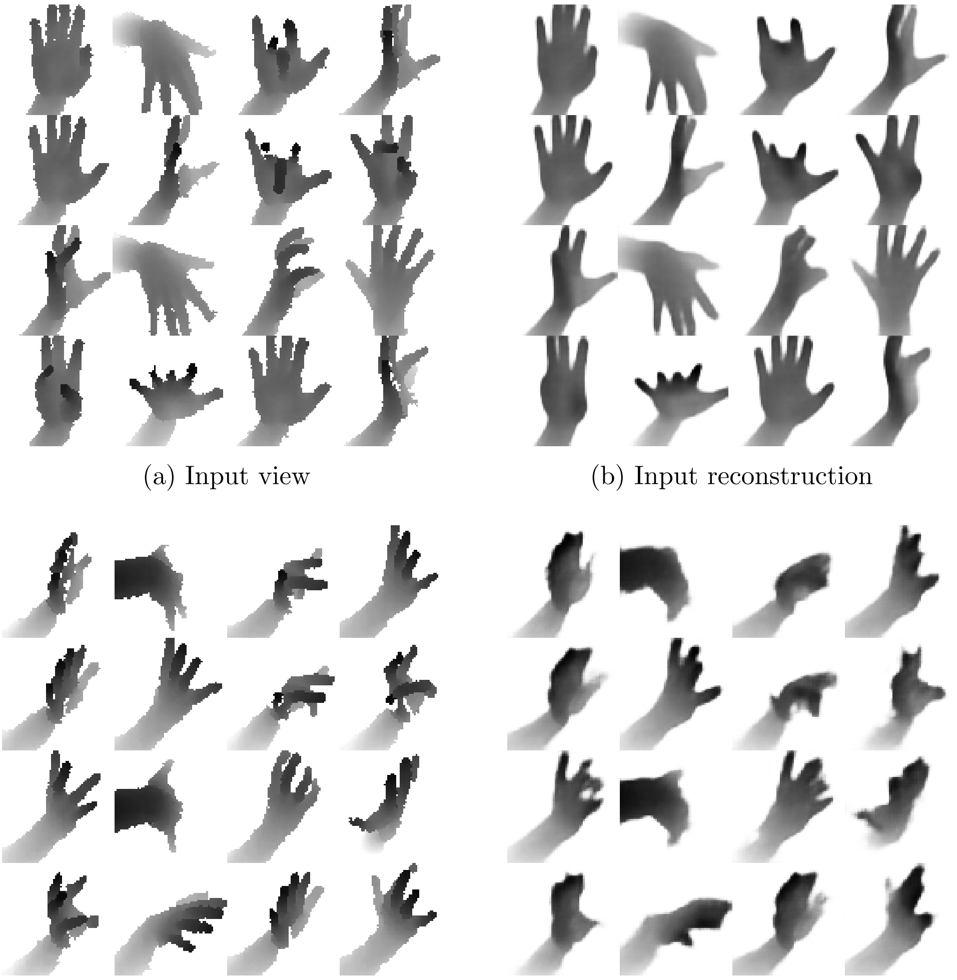
However, input reconstruction can be performed without knowledge about the pose, as the results in Table 4.1 indicate. Predicting different views, on the other hand, is a harder task but reveals pose information. These results suggest that our latent representation is predictive for the different view as well as the pose.

We believe that the reason for this large improvement in pose predictability is that our model is enforced to not just capture pixel statistics as can be sufficient to reconstruct the input (Hotelling, 1933; Kirby and Sirovich, 1990; Pearson, 1901). Enforcing the model to predict a different viewpoint requires the model to actually represent the pose. More specifically, our model needs to learn how the appearance affects the pose and thus the appearance in the other view.

Additional pre-training baseline A reviewer of the work (Poier et al., 2018) in which we first published the method described in this chapter, demanded Context Encoders (Pathak et al., 2016) as a more recent pre-training baseline. We briefly discuss this here. Context Encoders are an example of a strand of works on self-supervised learning, which split the input into parts and have the model learn about the relations between the parts (Doersch et al., 2015; Noroozi et al., 2017; Zhang et al., 2017). In particular, Context Encoders are trained to do inpainting. That is, large random contiguous parts of the input image are removed for training and the model should learn to inpaint the missing regions based on the context. The idea is that the model needs to learn to recognize the objects in the context in order to accomplish this task.

In Table 4.2 we compare the latent representations – pre-trained by different methods – based on their predictability for the pose (*c.f.* Table 4.1). The results show that the representations pre-trained using Context Encoders (Pathak et al., 2016) are even less predictive for the pose than the representations learned by autoencoders.

One issue of the Context Encoder baseline is the domain gap between training and testing as has been discussed in (Zhang et al., 2017) and Chapter 2.3. That is, at training time parts of the input are missing, while the model is applied to full images at test time. Moreover, we believe that the main idea of Context Encoders does not really apply to pose estimation of articulated objects, where one part of the object does not necessarily contain



(a) Input view

(b) Input reconstruction

(c) Different view

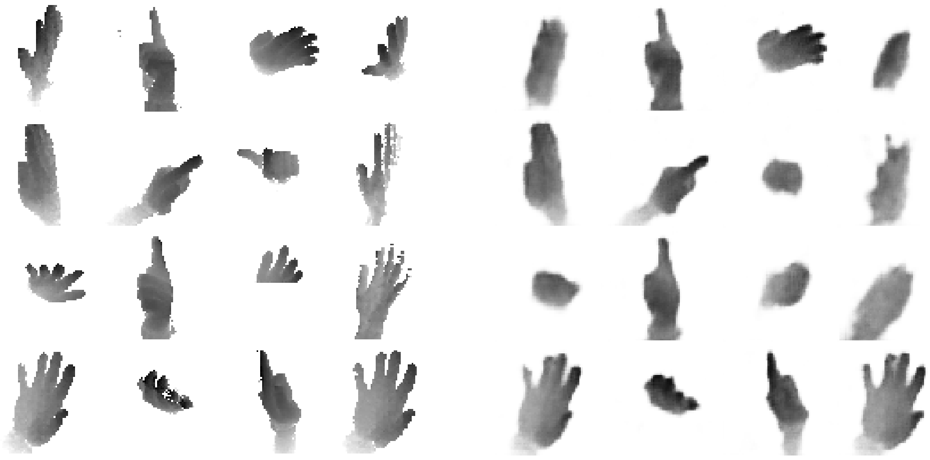
(d) Prediction for different view

Figure 4.3: **Input reconstruction vs. different view prediction.** Examples for generated views from the NYU validation set. Input view (a), reconstructions generated by the autoencoder (b), images from a different view (c), and the corresponding predictions from our method (d). Images from (a)-(d) with same grid index are corresponding. Visually, the autoencoder's input reconstructions resemble the input more closely than the predictions of our method match the different view. However, the latent representation learned by our method is much more predictive for the pose (*c.f.*, Table 4.1).

pose information for other parts. Hence, the learned latent representation contains little pose related information.

Size of the latent representation

We expect the size of the latent representation to be an important constraint for the specificity of the learned pose representation (*c.f.* (Oberweger et al.,



(a) Different view

(b) Prediction for different view

Figure 4.4: **View prediction examples on MV-hands data.** Target view (a), *i.e.*, ground truth images of the different view and the corresponding predictions from our method (b). Images with same grid index are corresponding.

Table 4.2: **Comparison of different pre-training methods on the NYU-CS dataset.** Mean joint error for learning a linear layer on top of the frozen latent representation with different numbers of labeled samples n. Best results in boldface.

Number of samples	100	1,000	10,000	43,640
Context Encoders (Pathak et al., 2016)	53.4	53.4	53.3	53.8
Autoencoder	48.0	47.2	47.3	47.1
PreView (Ours)	**33.4**	**29.6**	**29.0**	**29.0**

2015a; Tekin et al., 2016)). Hence, we investigate how the size of the representation affects the results, *i.e.*, the predictability of the pose. For this hyperparameter evaluation we employ the NYU validation set. We compare the results for representations of size $d_\mathcal{T} \in \{10, 20, 30, 40, 50, 80\}$ in Figure 4.5. It shows that the mean joint error is reduced by a large margin when increasing $d_\mathcal{T}$ from 20 to 30, but the improvement diminishes if $d_\mathcal{T} \sim 40$. It seems that, when trained in the proposed way, a size of 20 and below is too small to capture the pose and shape parameters reasonably well. However, if the size of the representation is increased above 50 the predictability of the pose is not improved anymore. This is interesting, since the size of the parameter space, which was identified by works on hand modeling (Albrecht et al., 2003; Lin et al., 2000) is usually very similar. The size identified in these works is indeed slightly smaller when representing the pose alone. Reasons for this might include that in our case the learned

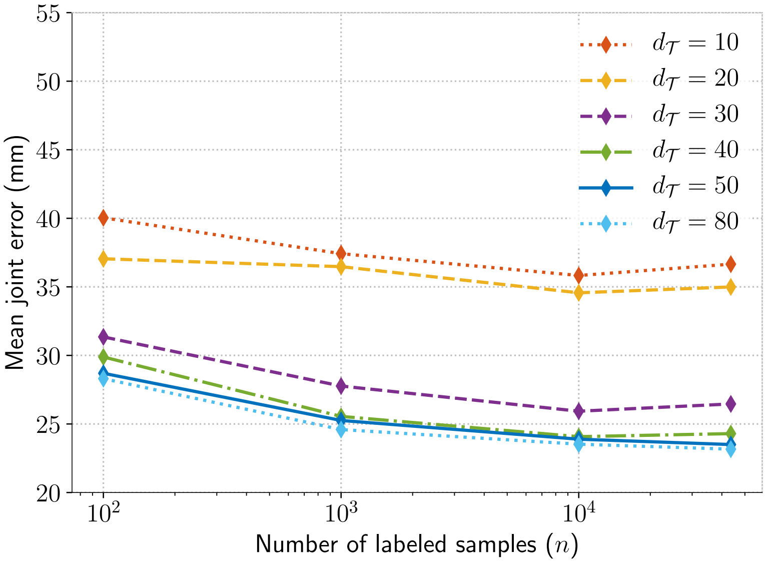


Figure 4.5: **Pose predictability.** How the size of the latent representation, $d_\mathcal{T}$, affects the predictability of the pose (from pre-trained, frozen representations). Results on the NYU validation set.

latent representation also needs to capture the size and shape of the hand or that the linear mapping to the joint positions based on which we perform this evaluation is learned independently from the latent representation and is thus unable to map to the joint positions for a very low dimensional latent representation.

Neuron activations

In another experiment we aim to qualitatively investigate what each neuron in the latent space has learned. To do this we search for the samples from the validation set, which activate a single neuron most. Figure 4.6 shows these samples for each neuron. We find that many of the neurons are activated most for very specific poses. That is, the samples, which activate a neuron most, clearly show similar poses.

Nearest neighbors

To obtain further insights into the learned representation, we visualize nearest neighbors in the latent representation space. More specifically, given a query image from the validation set, we find the closest samples from the training set according to the Euclidean distance in the latent representation space. Figure 4.7b visualizes some randomly sampled query images (*i.e.*, no "cherry picking“) and their corresponding nearest neighbors. We see that

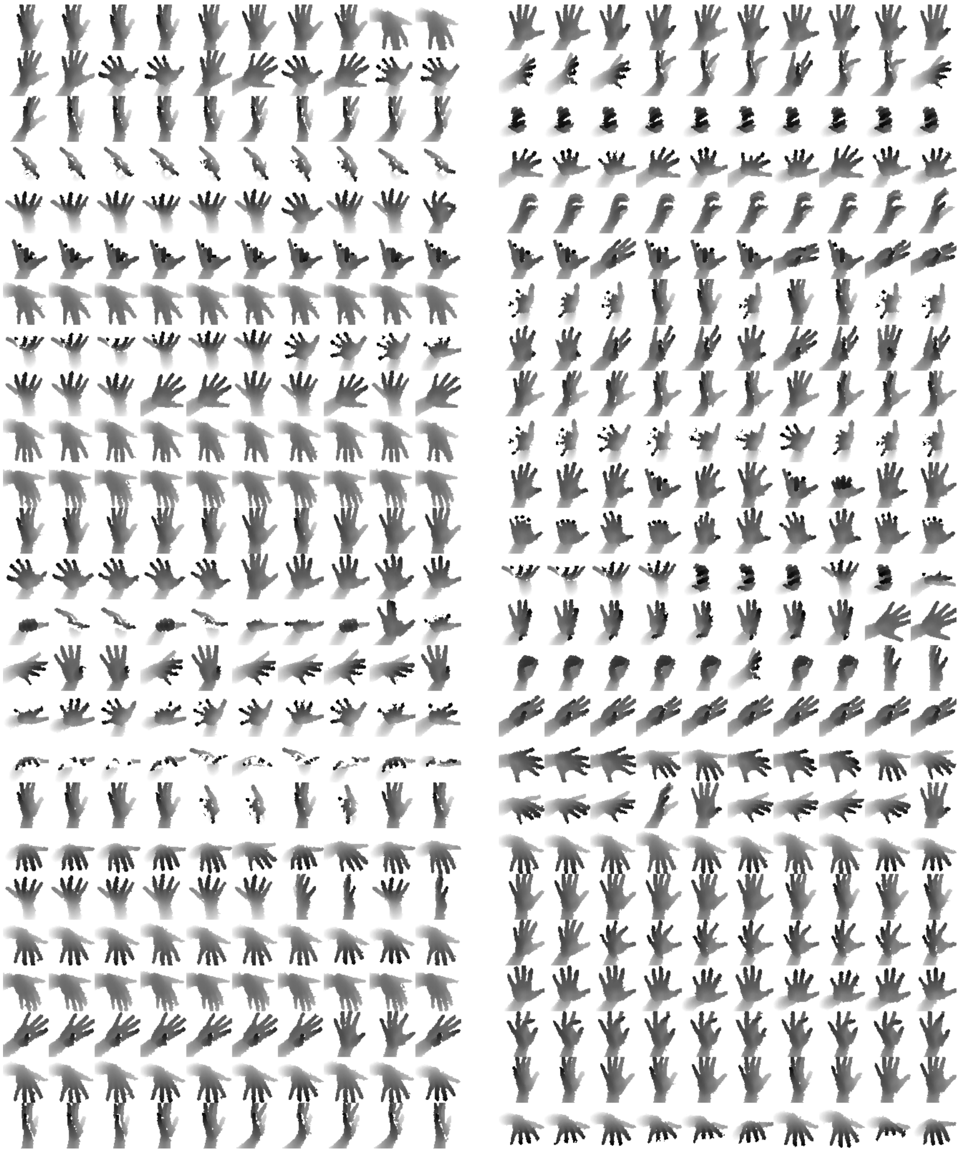
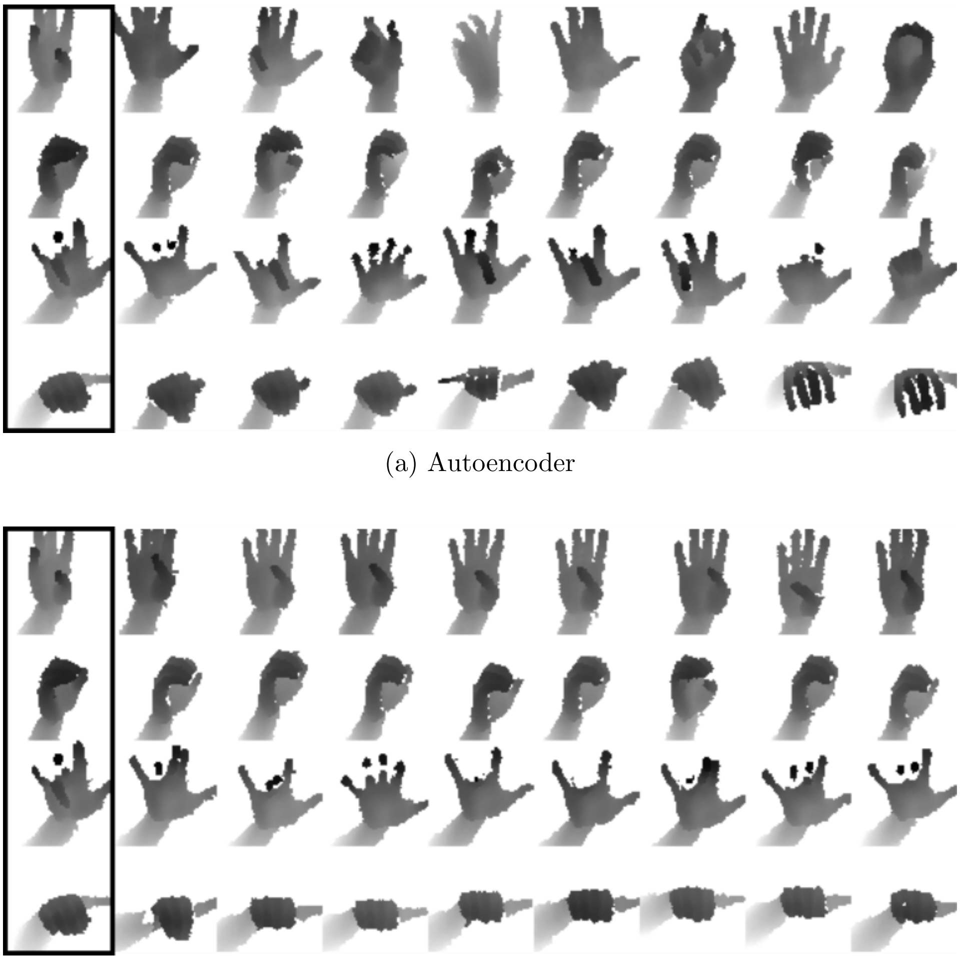


Figure 4.6: **Most activating samples.** Each row on the left and right side shows the ten samples from the validation set, which activate one neuron in the learned latent representation most. Note, that we randomly perturbed detections to verify the robustness of our method. Hence, sometimes parts of the hand are cut off in the crops.



(a) Autoencoder

(b) PreView (Ours)

Figure 4.7: **Nearest neighbors in latent space.** Comparison of nearest neighbors in the latent representation space for representations learned using an autoencoder (a) and our method (b). Query images (same queries shown for both methods) – randomly sampled from the validation set – are shown in the marked, leftmost column of (a) and (b), the remaining columns are the respective nearest neighbors.

the nearest neighbors most often exhibit a very similar pose as the query image, even if the detection (*i.e.*, hand crop) is not always accurate. This is in contrast to the nearest neighbors in the latent representation learned using autoencoders, which often show a completely different pose (see Figure 4.7a).

Similarly in Figure 4.8, we show nearest neighbors for the MV-hands dataset. Again, the nearest neighbors in the latent space exhibit very similar poses.

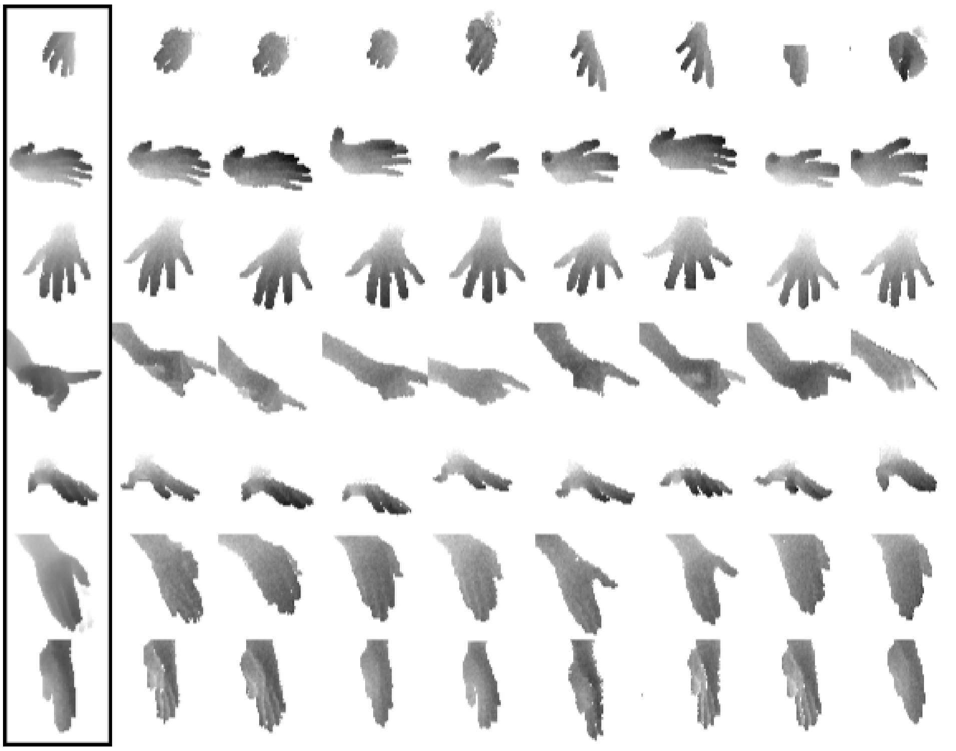


Figure 4.8: **Nearest neighbors in latent space.** Nearest neighbors from training set of the MV-hands dataset for query samples from validation set. Query images are shown in the marked, leftmost column, the remaining eight columns are the respective nearest neighbors.

4.4.4 Semi-supervised training

In a final set of experiments we test the proposed method for jointly leveraging labeled and unlabeled data (*c.f.* Sec. 4.3.1) during end-to-end training. Similar to the previous setup, we consider the case where the number of labeled samples is smaller or equal than the number of unlabeled samples, and evaluate different ratios. For a small number of labeled samples we obtained the best results by sampling the mini-batches such that there is an equal amount of labeled and unlabeled samples in each batch (*c.f.*, (Zhou et al., 2017b)).

Comparison to the state-of-the-art

To evaluate the competitiveness of the employed architecture, we compare against the state-of-the-art in data-driven hand pose estimation. Since the NYU-CS set contains about 60% of the original training set, we need to re-train the state-of-the-art approaches on the same subset for a fair comparison. We compare to Crossing Nets (Wan et al., 2017), DeepPrior (Oberweger et al., 2015a) and DeepPrior++ (Oberweger and Lepetit, 2017). We selected DeepPrior, since its results are still in the range of the state-of-the-art for the NYU dataset (as shown in a recent independent evaluation (Yuan et al., 2017)), the PCA based "prior" makes the approach suffer less from a reduced training set, and finally, it has about the same number of model parameters as our model. The improved variant DeepPrior++, on the other hand, has very recently been shown to be top-performing on different datasets (Oberweger and Lepetit, 2017).

To train the state-of-the-art approaches, we use the publicly available source code provided by the authors. Note, Wan et al. (2017) used different models for the experiments on the NYU dataset than the ones used in their publicly available code. For a fair comparison we use the same (metric) crop size when cropping the hand for the entire training and test set, and fix the training and validation subsets to the same subsets as for the evaluation of our method.

The results in Table 4.3 and 4.4 show that – by leveraging unlabeled data – our method consistently improves the performance, independent of the number of labeled samples, and improves the state-of-the-art approaches by a large margin for a small number of labeled samples. Note that the NYU dataset does not provide additional unlabeled samples, *i.e.*, when all labeled samples are used, our method can not draw from any additional information.

Ablation experiments

Finally, we focus the quantitative evaluation on the main contribution of this chapter. We exclude disturbing factors like the model architecture or the training procedure by training a baseline for which we keep everything the same except that we do not exploit unlabeled data and only train with labeled data.

In Table 4.4 we compare the results on the MV-hands dataset. We see that our semi-supervised training improves the results of supervised training for all metrics. Figure 4.9 compares the results on the NYU-CS dataset, where our method (*Semi-superv.*) also improves results for a high number of labeled samples. We also compare to the variant with the additional adversarial term (*Semi-superv. & Adversarial*; *c.f.*, Eq. (4.9)). We see that the additional adversarial term and corresponding training procedure can improve the results slightly for larger numbers of labeled samples n, but not in cases where only a small number of samples is labeled. Moreover, note that we obtained the presented results for the adversarial training by tuning hyperparameters separately for different n and taking the best results. We found that, for different n, different conditioning types and

Table 4.3: **Comparison to the state-of-the-art.** Results on the NYU-CS dataset for different metrics and different numbers of labeled samples n. For the mean joint error (ME) smaller values are better, while for the success rates (FS80 and JS80) higher values are better. Best results in boldface.

(a) $n = 100$

	ME	FS80	JS80
DeepPrior (Oberweger et al., 2015a)	44.99	0.11	0.45
Crossing Nets (Wan et al., 2017)	67.65	0.00	0.25
DeepPrior++ (Oberweger and Lepetit, 2017)	38.07	0.14	0.53
Semi-superv. Autoenc.	31.58	0.27	0.60
Semi-superv. PreView (Ours)	**29.12**	0.31	**0.63**
Semi-superv. & Adversarial (Ours)	29.52	**0.32**	**0.63**

(b) n=1,000

	ME	FS80	JS80
DeepPrior (Oberweger et al., 2015a)	36.99	0.20	0.55
Crossing Nets (Wan et al., 2017)	36.35	0.16	0.55
DeepPrior++ (Oberweger and Lepetit, 2017)	31.01	0.23	0.61
Semi-superv. Autoenc.	24.05	0.41	0.70
Semi-superv. PreView (Ours)	**22.96**	**0.44**	**0.71**
Semi-superv. & Adversarial (Ours)	23.32	0.41	0.70

(c) n=10,000

	ME	FS80	JS80
DeepPrior (Oberweger et al., 2015a)	30.31	0.31	0.63
Crossing Nets (Wan et al., 2017)	28.97	0.29	0.64
DeepPrior++ (Oberweger and Lepetit, 2017)	24.14	0.37	0.69
Semi-superv. Autoenc.	21.32	0.47	0.73
Semi-superv. PreView (Ours)	21.49	0.47	0.73
Semi-superv. & Adversarial (Ours)	**20.67**	**0.48**	**0.74**

(d) n=43,640

	ME	FS80	JS80
DeepPrior (Oberweger et al., 2015a)	27.97	0.35	0.66
Crossing Nets (Wan et al., 2017)	25.57	0.34	0.68
DeepPrior++ (Oberweger and Lepetit, 2017)	20.87	0.44	0.73
Semi-superv. Autoenc.	20.74	**0.49**	**0.74**
Semi-superv. PreView (Ours)	20.70	0.48	**0.74**
Semi-superv. & Adversarial (Ours)	**20.23**	**0.49**	**0.74**

Table 4.4: **Comparison to the state-of-the-art and ablation experiments.** Results for different metrics on the MV-hands dataset.

n	289		
Metric (see Sec. 4.4.2)	ME	FS80	JS80
DeepPrior++ (Oberweger and Lepetit, 2017)	34.17	0.22	0.57
Supervised	26.35	0.36	0.67
Semi-superv. Autoencoder	25.20	0.38	0.68
Semi-superv. PreView (Ours)	**24.14**	**0.39**	**0.69**

settings for λ_a worked best. While for a small number of labeled samples, $n = 100$, conditioning the discriminator solely on the input and a small weight for the adversarial term ($\lambda_\mathrm{a} = 0.01$) yielded best results, for larger n, conditioning on the pose and a larger weight $\lambda_\mathrm{a} = 0.1$ had a positive impact on the results.

The results point out that training our method with an additional adversarial loss term bears potential to improve results. However, it appears that a sufficient amount of labeled samples is necessary so that the discriminator can exploit the pose conditioning and provide improved feedback for training the decoder. Additionally, it requires significant tuning to achieve improved performance. The semi-supervised training without the adversarial term, on the other hand, does not require similar extensive hyperparameter tuning.

Overall, semi-supervised training consistently outperforms supervised training, even if all samples are labeled. For the more realistic case, where only a subset of the data is labeled, our method improves the performance of the fully supervised approach by a large margin. In fact, our method achieves similar or improved results even when it is trained with one to two orders of magnitude less labeled samples.

4.5 Conclusion

Learning from unlabeled data has long been recognized as an important direction for machine learning and is especially desirable for tasks with high labeling effort, such as estimation of articulated poses. However, traditionally the representations learned from unlabeled data are most often generic. While in this way the representations are amenable for transfer learning to novel tasks, concrete applications benefit from task specific representations.

In this chapter, we showed a way how to learn task specific representations for pose estimation without labels and that such task specific repre-

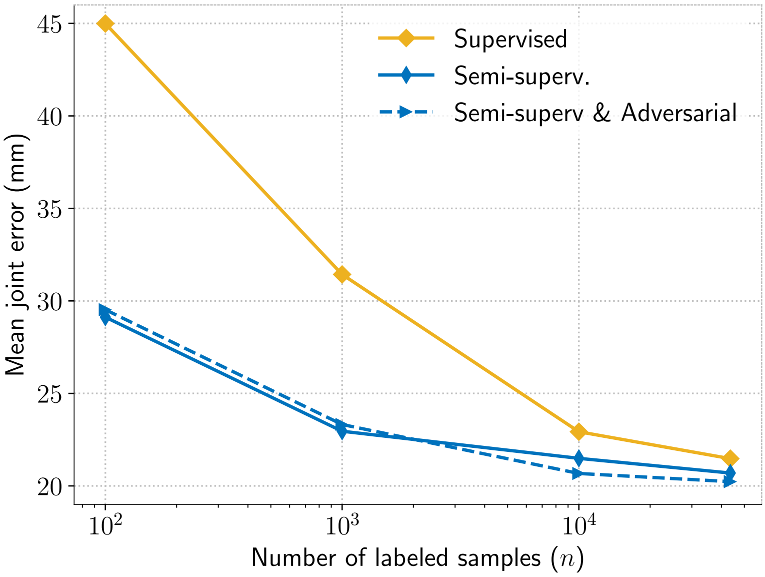


Figure 4.9: **Ablation experiments.** Comparison of purely supervised training (*Supervised*), with the proposed method which can exploit unlabeled samples (*Semi-superv.*) and the variant trained with the additional adversarial term (*Semi-superv. & Adversarial*) for different numbers of labeled samples n on the NYU-CS dataset.

sentation are beneficial compared to more generic ones. Additionally, the proposed method can be trained end-to-end in a semi-supervised manner. Our method consistently surpasses the performance of standard supervised training, even when all available training samples are labeled. Moreover, the results of supervised training are already improved with one order of magnitude less labeled training samples.

While a pose specific latent representation can be learned without requiring labeled data, in order to output a manually defined target representation (*e.g.*, joint positions), labeled data is still required. That is, labeled data is needed to learn the final mapping to the target space. Moreover, we see that this final mapping still yields better accuracy, the more labeled samples we provide.

Synthetic data opens up a way to provide labeled data without causing a significantly increased manual effort. Nevertheless, the exploitation of synthetic data is hampered by the domain gap between synthetic and real data. Building upon the method introduced in this chapter, in the next chapter we introduce a way to exploit synthetic data together with unlabeled data in order to mitigate the domain gap. We can thus learn the mapping to the target space from mainly synthetic data and ultimately a strong overall model from a very low number of labeled real samples.

CHAPTER 5

Exploiting synthetic data

Contents

Recent methods aiming to reduce the labeling effort for learning pose estimation models often employ synthetic data or semi-supervised learning (Neverova et al., 2017; Wan et al., 2017), which, both, have their specific drawbacks. Approaches, employing synthetic data have to deal with the domain gap, which has been recently approached for hand pose estimation by learning a mapping between the feature spaces of real and synthetic data (Rad et al., 2018b). Unfortunately, learning this mapping requires a large amount of labeled real data and corresponding synthetic data. On the other hand, semi-supervised approaches can better exploit a small amount of labeled data, however, the results are often still not competitive.

We aim to overcome these issues by exploiting accurately labeled synthetic data together with unlabeled real data in a specifically devised semi-supervised approach. We employ a large amount of synthetic data to learn an accurate pose predictor, and, inspired by recent work (Massa et al., 2016; Rad et al., 2018b), learn to map the features of real data to those of synthetic data to overcome the domain gap. However, in contrast to previous work, we learn this mapping mainly from unlabeled data.

We train the mapping from the features of real to those of synthetic data using two auxiliary objectives based on unlabeled data. One objective enforces the mapped features to be pose specific, and the other one enforces the feature distributions of real and synthetic data to be aligned.

For the first of the two auxiliary objectives, which is responsible for enforcing a pose specific representation, we build upon the idea described in Chapter 4. Therein we showed that by learning to predict a different view from the latent representation, the latent representations of similar poses are pushed close together. That is, the only necessary supervision to learn such a pose specific representation can be obtained by simply capturing the scene simultaneously from different view points. In this chapter we employ the same idea to enforce the joint latent representation of real and synthetic data (*i.e.*, after mapping) to be pose specific by enforcing the representation to be predictive for the appearance in another view.

The second objective is to align the feature distributions of real and synthetic data. The underlying idea of learning a mapping from the features of real samples to the features of synthetic samples is that the labeled synthetic data can be better exploited if real and synthetic samples with similar poses are close together in the latent space. Simply ensuring that the latent representation is pose specific does, however, not guarantee that the features of real and synthetic data are close together in the latent space: Similar poses could form clusters for real and synthetic data, *individually*. To avoid this, we employ an adversarial loss, which acts on the latent space and penalizes a mismatch of the feature distributions.

By simultaneously ensuring that similar poses are close together and feature distributions are aligned, we show that we are able to train state-of-the-art pose predictors – already with small amounts of labeled real data. More specifically, employing about 1% of the labeled real samples from the NYU dataset (Tompson et al., 2014) our method outperforms many recent state-of-the-art approaches, which use all labeled real samples. Furthermore, besides quantitative experiments, we perform qualitative analysis showing that the latent representations of real and synthetic samples are well aligned when using mainly unlabeled real data. Moreover, in our extensive ablation study we find that, both, enforcing pose specificity as well as aligning the distributions of real and synthetic samples benefits performance (see Sec. 5.3.3).

5.1 Related work

As discussed in this thesis, recent works on hand pose estimation heavily rely on data-driven approaches – either as a stand-alone approach (Keskin et al., 2012; Madadi et al., 2017; Tang et al., 2014), or to guide a model-based approach (Panteleris et al., 2018; Sharp et al., 2015; Ye et al., 2016). The effectiveness of such a data-driven approach crucially depends on how well the dataset it is trained on covers the pose space.

Training data and annotation Given the crucial role of annotated training data for state-of-the-art approaches to hand pose estimation, a lot of effort has been devoted to the creation of training data sets. Many semi- or fully-automatic approaches have been employed to label real data (Sun et al., 2015; Tompson et al., 2014; Yuan et al., 2017). Still, these are often difficult to set up, require a significant amount of manual interaction, and/or great care has to be taken to avoid that attached sensors affect the data too strongly. All these efforts point out that the development of methods which reduce the dependence on labeled real data would foster quicker deployment and make such systems more accessible.

Synthetic data One way to lessen the effort for labeling real data is to employ synthetic data. Synthetic data has the advantage that it has perfectly accurate labels and a virtually infinite number of samples can be generated. However, the data generating distribution usually differs between the synthetic training data and the real test data. Hence, models trained only on synthetic data suffer from the so-called *domain gap* and usually perform significantly worse than models trained on real data (Abdi et al., 2018; Rad et al., 2018b).

Unlabeled data and domain adaptation Besides synthetic data, unlabeled real data can be used to lessen the labeling effort as we have shown in the previous chapter (Chapter 4). In Chapter 4 we only employ real data and, hence, do not have to deal with a domain gap. However, for a small number of labeled and a large number of unlabeled data this approach alone will usually not be competitive due to the reduced pose supervision.

Other works try to boost performance by combining labeled synthetic and unlabeled real data (Abdi et al., 2018; Mueller et al., 2018; Shrivastava et al., 2017; Tang et al., 2013). To mitigate the domain gap between these two distributions they most often use a framework based on Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). For instance in (Liu and Mian, 2017; Mueller et al., 2018; Shrivastava et al., 2017) a model is learned to transform synthetic images to corresponding real images, which can then be used for training using the accurate labels from the initial synthetic data. Our method is orthogonal. We show how synthetic and unlabeled real data can be used to learn a pose specific latent representation, which can directly be used during inference.

Our method is closely related to approaches that aim to overcome the domain gap by learning a shared latent space for different modalities. For instance in (Spurr et al., 2018; Wan et al., 2017) a shared latent space is learned for images and poses. Similarly, Rad et al. (2018b) incorporate synthetic data by learning to map the features of real samples to the features of synthetic samples. Abdi et al. (2018) take the idea of a shared latent space further and incorporate poses, synthetic samples as well as labeled and unlabeled real samples. Similar to Wan et al. (2017) they combine a Variational Autoencoder (VAE) (Kingma and Welling, 2014) with a GAN and exploit unlabeled samples during training the GAN, which in turn improves the overall system. In contrast to their work, in our method the adversarial term does not operate on the images but directly on the much lower dimensional latent space, for which it should be easier to train a discriminator-generator pair of lower complexity. Moreover, we enforce pose specific constraints on the latent representation of both, labeled and unlabeled data, which yields a significant performance gain. For a more general discussion on transfer learning and domain adaptation see Chapter 2.4.

5.2 Mapping unlabeled real to synthetic data

Our method builds on the basic observation that for hand pose estimation from depth images it is easy to obtain labeled synthetic data and unlabeled real data. First, a large number of synthetic data is used to train a very strong pose predictor. To make this strong predictor amenable for real data, we learn to map real data to synthetic data. Crucially, our goal is to learn this mapping between real and synthetic data with as little ground truth supervision as possible. Instead, we learn the mapping by mainly relying on unlabeled data.

To properly exploit unlabeled together with synthetic data, we propose two auxiliary loss functions. The first one uses a self-supervised term, which enforces the joint latent representation of synthetic and real data to be pose specific without the need for pose labels by ensuring that the representation is predictive for the hands' appearance in another view. The second loss is an adversarial loss which ensures that the feature distributions for real and synthetic data are aligned. That is, we simultaneously ensure that the distributions are matched and the representation is pose specific. Ultimately, the training loss joins the target loss, ℓ_p, with an loss for matching corresponding real and synthetic samples ℓ_c for labeled data and the two

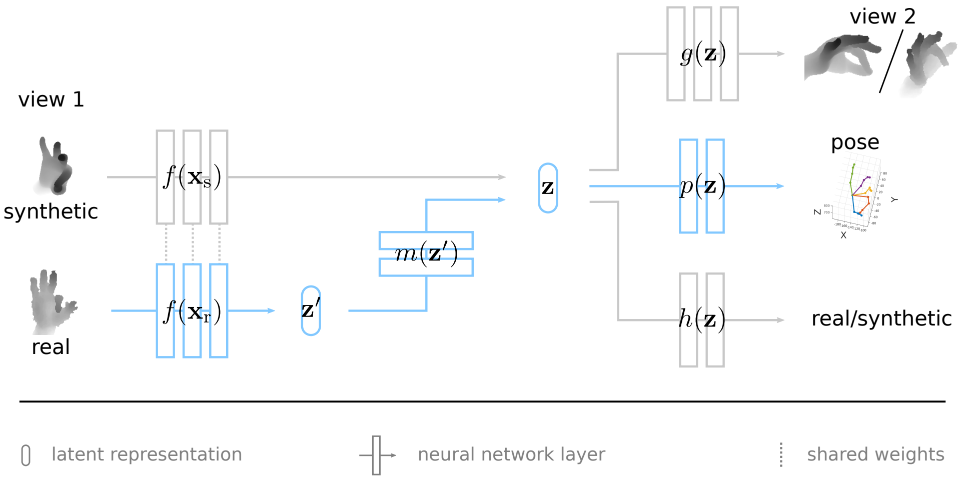


Figure 5.1: **Sketch of the architecture.** We train our system jointly with real and synthetic samples. From the joint latent representation $\mathbf{z}$, we predict the pose as well as two auxiliary outputs from which we also obtain feedback for unlabeled data. The auxiliary objectives are (i) to predict a different view and (ii) to discriminate between real and synthetic data. The training using unlabeled data ensures aligned latent feature distributions and thus, improved exploitation of synthetic data even with a small amount of labeled real samples. During test time only the pose is predicted (blue path). Note that the layers per module are just for illustration and do not represent the actual number of layers.

auxiliary losses ℓ_g and ℓ_m, which also base on unlabeled data:

$$\ell = \ell_p + \lambda_c \ell_c + \lambda_g \ell_g + \lambda_m \ell_m, \tag{5.1}$$

where ℓ_g is the loss of the self-supervised term, ℓ_m is the adversarial loss to match the feature distributions, and λ_c, λ_g and λ_m are respective weighting terms. Figure 5.1 depicts the overall architecture of our method, giving rise to the individual loss terms. In the following we describe all terms in detail.

5.2.1 Predicting the pose

For the description of our method we assume the learned model employed for pose prediction to be based upon two separate functions. A function f, which transforms the input to some latent space and a second function p, which maps from the latent space to the desired target space. That is, given an input image $\mathbf{x}$, the function f will produce a latent representation,

$$\mathbf{z} = f(\mathbf{x}). \tag{5.2}$$

The function p, on the other hand, maps a given latent representation $\mathbf{z}$ to a pose representation,

$$\hat{\mathbf{y}} = p(\mathbf{z}), \tag{5.3}$$

where the target space can be any pose representation (*e.g.*, joint positions). Hence – successively applied – these two functions map the input image to a pose representation:

$$\hat{\mathbf{y}} = p(f(\mathbf{x})). \tag{5.4}$$

In our work we implemented these two functions as neural networks. Similar to other works (Rad et al., 2018b; Wan et al., 2017), we train the networks to directly output 3D joint positions. For comparability to baselines we employ the mean squared error to learn the network parameters. That is, the target loss is simply the squared L^2-norm:

$$\ell_p = \sum_k \|\mathbf{y}_k - \hat{\mathbf{y}}_k\|_2^2 \,, \tag{5.5}$$

where $\mathbf{y}_k$ is the ground truth and $\hat{\mathbf{y}}_k$ is the prediction for the k-th sample.

5.2.2 Mapping real to synthetic data

To train a neural network for hand pose estimation we can generate a virtually infinite amount of synthetic data. The model trained solely on synthetic data will, however, not work similarly well on real images (*e.g.* *c.f.*, quantitative results in Sec. 5.3.3). To see why this happens we visualize the feature distribution of real and synthetic data from a model trained solely with synthetic data in Figure 5.2. The visualization indicates that the features of corresponding real and synthetic samples take up different areas in the feature space, *i.e.*, they are not aligned. And hence it will not be straightforward for a model trained only with synthetic data to make the same prediction for a synthetic sample and a real sample with the same pose, as the features of real samples are often not close to their accurately corresponding synthetic samples.

To overcome this problem, we take inspiration from recent works (Massa et al., 2016; Rad et al., 2018b), which learn to map the features of real images to the features of synthetic images. In this way, a large amount of synthetic images can be exploited to train a strong pose prediction model, which then – after mapping the features – also yields improved performance on real images.

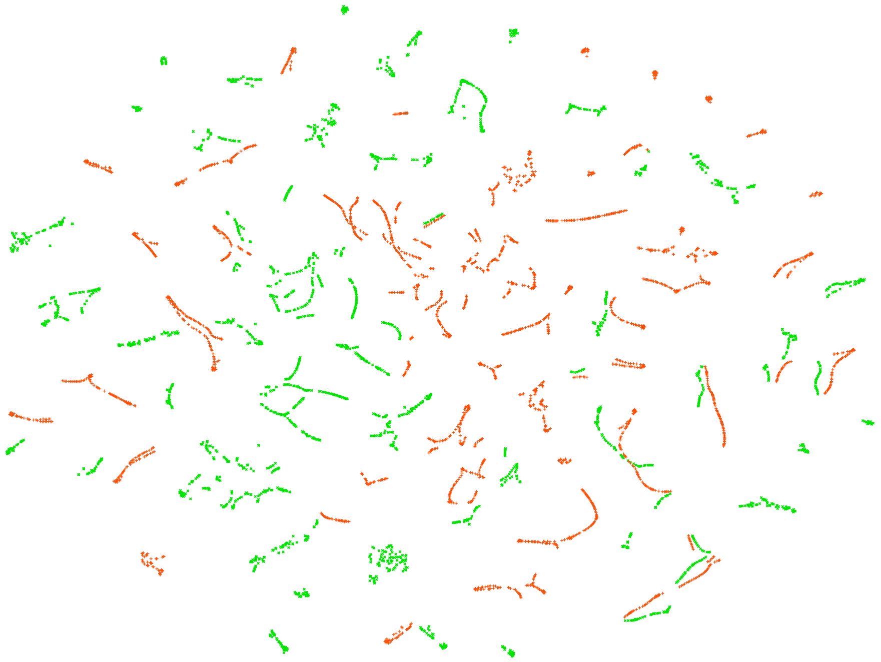


Figure 5.2: **Visualization of latent representations.** t-SNE visualization (van der Maaten and Hinton, 2008) of the latent representations for corresponding real (green; ✖) and synthetic (orange; ✚) validation samples when the model was trained only using synthetic data. While real and synthetic samples are corresponding, *i.e.*, they exhibit the same poses, they are not aligned in the feature space. For a similar visualization for a model trained with our system *c.f.* Figure 5.9.

More specifically, a function m is trained to map from the features $\mathbf{z}'$ of a real image to the feature space of synthetic images:

$$\hat{\mathbf{z}} = m(\mathbf{z}'), \tag{5.6}$$

where $\hat{\mathbf{z}}$ denotes the latent representation of a real image in the feature space of synthetic images. Hence, employing the whole model to predict the pose of the hand in a real image, $\mathbf{x}_\mathrm{r}$, we successively apply functions f to extract features, m to map the features and, finally, p to predict the pose:

$$\hat{\mathbf{y}} = p(m(f(\mathbf{x}_\mathrm{r}))). \tag{5.7}$$

To learn the mapping function m, Massa et al. (2016) and Rad et al. (2018b) require a one-to-one correspondence between real and synthetic data. The mapping is then trained to minimize the distance between the mapped feature representation $\hat{\mathbf{z}}$ of a real image and the feature representation $\mathbf{z}$ of the corresponding synthetic image. For the available corresponding real and synthetic samples we follow this approach and aim to minimize the squared L^2-norm of the distance between corresponding feature representations. That is,

$$\ell_c = \sum_{k \in \mathcal{C}} \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_2^2\,, \tag{5.8}$$

where $\mathcal{C}$ denotes the set of available corresponding real and synthetic samples. Employing the squared L^2-norm assumes rather accurately corresponding real and synthetic samples. Such corresponding samples are available for our evaluation datasets but can be a large effort to provide in general. This is because finding a synthetic image, which accurately corresponds to a given real image in terms of the pose is indeed equivalent to labeling the pose of the real image. While the choice of the norm depends on the accuracy and reliability of the correspondences, independent of the choice of the norm, a large number of corresponding samples is required to learn the mapping between the feature distributions. Hence, relying solely on this approach would still require to have a significant amount of labeled real images. In this chapter we introduce ways to overcome this requirement and reduce the number of necessary corresponding real and synthetic images. We describe them in the following.

5.2.3 Learning to map from unlabeled data

We aim to train the mapping (Eq. (5.6)) without requiring a large amount of labeled real samples. To do this we add two auxiliary loss functions exploiting unlabeled data to train the mapping. One of them enforces the mapped representation to resemble the pose, for both, real and synthetic data. At the same time, a second loss ensures that the model does not push the features of real and synthetic images apart, *i.e.*, ensure that the feature distributions are aligned. Together, these two auxiliary loss functions enable us to effectively train the mapping from mainly unlabeled samples.

5.2.3.1 Learning pose specifity from unlabeled data

We want to map representations of images showing a similar pose close together. Since we do not have labels we cannot enforce this directly. In the previous chapter, however, we introduced a way to enforce this indirectly – solely based on unlabeled data. Here we build upon this idea. We train a decoder, which is given the latent representation and trained to predict a second view of the hand. Recall that the idea is that, if the decoder is able to predict another view of the hand solely from the latent representation, the latent representation must contain pose specific information.

That is, we train a decoder g, which – given a pose specific feature representation $\mathbf{z}$ – is able to predict the hands' appearance $\mathbf{x}^{(j)}$ from a different view j, *i.e.*,

$$\hat{\mathbf{x}}^{(j)} = g(\mathbf{z}^{(i)}), \tag{5.9}$$

where $\mathbf{z}^{(i)}$ denotes the feature representation produced for an input from view i.

To train such a generator function g we do not need any labels, we only need to capture the hand simultaneously from a different viewpoint or render the hand model from a different virtual view for synthetic samples, respectively. That is, the objective for the generator – given only the latent representation – is to predict the appearance of the hand as captured/rendered from the second view. Hence, to train g we employ the same reconstruction loss as described in the previous chapter:

$$\ell_g = \sum_k \left\| \mathbf{x}_k^{(j)} - \hat{\mathbf{x}}_k^{(j)} \right\|_1, \tag{5.10}$$

where $\mathbf{x}_k^{(j)}$ is the captured/rendered image and $\hat{\mathbf{x}}_k^{(j)}$ is the model prediction for the k-th image from view j.

In the previous chapter we showed that only using the view prediction objective, the latent representation can be enforced to be pose specific without the need for any pose labels. Nevertheless, for our case we do not have corresponding real and synthetic data. That is, given a synthetic sample the target for the generator g is a synthetic sample and, equivalently, for a real sample the target for the generator is a real sample. In this way the generator g – besides trying to generate the correct appearance corresponding to the pose of the sample – might also try to discriminate between real and synthetic samples in order to accurately predict the appearance. This would clearly counteract the goal of learning a shared latent space, where real and synthetic samples with similar poses are close together. In the next section we show how we overcome this issue.

5.2.3.2 Matching feature distributions from unlabeled data

Enforcing the latent representation $\mathbf{z}$ to be specific for the pose does not ensure that real and synthetic samples with similar poses are mapped to similar latent representations. Indeed, real and synthetic samples could be pushed into different areas of the feature space by the non-linear functions f and m, respectively. Such a separation in the feature space would clearly hamper the exploitation of synthetic data for training a pose predictor for real data.

To avoid a scenario where the latent representations of real and synthetic samples are pose specific but still separated in the feature space, we need a way to ensure that similar poses are mapped to similar latent representations, independently of whether the samples are real or synthetic. Without having corresponding real and synthetic samples, this is difficult to ensure on the level of individual samples. However, as long as we can assume that the distribution of poses is similar for real and synthetic data, we can enforce that also the feature distributions match. By ensuring that the feature distributions match, and at the same time ensuring that the features are pose specific, similar poses should yield similar pose representations for, both, real and synthetic samples, which was the initial goal.

Here, we enforce the feature distributions of real and synthetic data to match by employing an adversarial training loss (Goodfellow et al., 2014). The adversarial loss operates on the latent representations, *i.e.*, we use a discriminator, which is trained to discern real and synthetic samples given the latent representation. The mapping function m, on the other hand, should make the latent representation of real samples as similar as possible to the latent representation of synthetic samples and, hence, indiscernible for the discriminator.

For the implementation we follow the formulation of Least Squares GAN (Mao et al., 2017), which has shown to work well for adversarial training. In this formulation the discriminator function h predicts a real-valued label:

$$\hat{l} = h(\mathbf{z}), \quad \hat{l} \in \mathbb{R}, \tag{5.11}$$

which should be $l_{\mathrm{r}} = 1$ for real and $l_{\mathrm{s}} = 0$ for synthetic samples, respectively. Consequently, the loss for the discriminator penalizes deviations from these target values for predictions on respective samples:

$$\ell_h = \frac{1}{2} \sum_{k \in \mathcal{R}} \left(\hat{l}_k - l_{\mathrm{r}} \right)^2 + \frac{1}{2} \sum_{k \in \mathcal{S}} \left(\hat{l}_k - l_{\mathrm{s}} \right)^2, \tag{5.12}$$

where $\mathcal{R}$ is the set of real, and $\mathcal{S}$ the set of synthetic samples, respectively.

The loss for the mapping function m, on the other hand, enforces real samples to be indiscernible from synthetic samples for the discriminator:

$$\ell_m = \frac{1}{2} \sum_{k \in \mathcal{R}} \left(\hat{l}_k - l_{\mathrm{s}} \right)^2. \tag{5.13}$$

We choose the least squares objective here as it overcomes problems with vanishing gradients compared to the original GAN formulation (Goodfellow et al., 2014). The original formulation minimizes the Jensen-Shannon divergence and hence employs a classification loss. Using a classification loss, the gradients for the mapping function may vanish during training when real samples are classified as synthetic by the discriminator (which would be the goal for the mapping function) but the real samples are still out of the distribution of the actual synthetic samples (Arjovsky et al., 2017; Mao et al., 2017). While, in such a case, the mapping function would not receive a significant update with a classification loss, the least squares objective provides usable gradients everywhere. This appears especially interesting for our task since we are not targeting a classification but a regression task.

Our analysis in Sec. 5.3.5 shows that using the adversarial training the latent representations of real and synthetic samples can be well aligned from mainly unlabeled samples. More importantly, we find that in this way the pose estimation results are improved.

5.3 Experiments

To verify the applicability of our method we compare the results to recent semi-supervised and fully supervised state-of-the-art methods. Furthermore, we investigate the contribution of the individual parts of our method in an ablation study, inspect the learned latent representation to obtain more insights in how it is affected by our contributions and analyze failure cases of our method.

5.3.1 Experimental setup

Here we describe the details of the experimental setup employed to evaluate the system introduced in this chapter. Additionally, we make the implementation of our method publicly available[1].

As in many recent works we crop a square region around the hand location, resize it to a 128×128 patch and normalize the depth values to the range $[-1, 1]$ (Abdi et al., 2018; Oberweger and Lepetit, 2017; Rad et al., 2018b). These patches are fed into the network. The batch size for training is 64. We pre-train f and p with synthetic data for about 170k iterations and subsequently train the whole model (Eq. (5.1)) jointly with real and synthetic data for about 140k iterations.

Architecture In principle, our contribution is agnostic to the network architecture. To verify our method we adopt architectures from recent work for the individual modules (f, m, g, h and p; *c.f.* Figure 5.1) of our system. The feature extractor f is similar to the model used in (Oberweger and

[1] `https://poier.github.io/murauer`

Lepetit, 2017), *i.e.*, an initial convolutional layer with 32 filters of size 5 × 5 is followed by a 2 × 2 max-pooling, four "residual modules" with 64, 128, 256, and 256 filters, respectively, each with five residual blocks (He et al., 2016), and a final fully connected layer with 1024 output units. The pose estimator p consists of two fully connected layers, with 1024 and $3J$ outputs, respectively, where J is the number of predicted joint positions in our case. The mapping layer m is adopted from (Rad et al., 2018b), *i.e.*, it consists of two residual blocks, each with 1024 units. The discriminator h has the same architecture as the mapping m with an additional linear layer to predict a single output. As in the previous chapter, the generator g is based on the generator of DCGAN (Radford et al., 2016). It consists of four layers of transposed convolutions, each followed by Batch Normalization (Ioffe and Szegedy, 2015) and a leaky ReLU activation (Maas et al., 2013). We add a bilinear upsampling layer prior to the final hyperbolic tangent (tanh) activation in order to upsample from 64 × 64 to 128 × 128 in our case.

Optimization For optimization of the model parameters we use Adam (Kingma and Ba, 2015) with standard parameters, *i.e.*, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also found it helpful to follow a *warm-up* scheme for the learning rate and decay the learning rate gradually later (Goyal et al., 2017). More specifically, we start with about one tenth of the pre-defined learning rate α_0 and approximately triple it after the first epoch. We start training with the pre-defined learning rate α_0 – subject to exponential decay – after three epochs. That is, the learning rate α_e for epoch e is computed by:

$$\alpha_e = \eta_e \, \alpha_0, \tag{5.14}$$

with the scaling factor

$$\eta_e = \begin{cases} 0.33^{2-\lfloor \frac{e}{2} \rfloor} & \text{if } e < 4 \\ \exp(-\gamma \, e) & \text{otherwise,} \end{cases} \tag{5.15}$$

where γ determines the speed of the decay and is set to 0.04 in our case. In our experiments $\alpha_0 = 3.3 \times 10^{-4}$ yielded the best results. Here, the notion of epoch is always based on the number of real data samples in the dataset (72,757 for the NYU dataset) and independent of the actually used dataset (*e.g.*, sub-sampled real data, synthetic data, *etc.*). That is, the number of iterations per epoch is the same for all experiments (1,137 iterations per epoch with a batch size of 64).

Loss weights λ and mini-batch sampling We found the loss weights used in our evaluation experimentally and set $\lambda_c = 0.2$, $\lambda_g = 10^{-4}$ and $\lambda_m = 10^{-5}$. For each mini-batch we independently sample a set of corresponding real and synthetic samples, independent sets of real and synthetic samples and a set of unlabeled samples such that there is an equal number of samples from each of the four sets (*i.e.*, 16 samples per set in our case).

Dataset and metric Again, we employ the NYU hand pose dataset (Tompson et al., 2014), since it is the single prominent dataset providing data captured from multiple view points together with synthetic data, which we can readily use to compare to the results of state-of-the-art approaches. This dataset was captured with three RGBD cameras simultaneously. It contains 72,757 frames for training and 8,252 frames for testing. In some works a subset of 2,440 samples from the test set is used as a validation set (Wan et al., 2017). We use this set for analyzing the latent space in Sec. 5.3.5. Following standard convention we evaluate on 14 joints (Ge et al., 2018; Moon et al., 2018; Tompson et al., 2014) using the commonly used mean joint error (ME) (Moon et al., 2018; Oikonomidis et al., 2011b; Sun et al., 2015). The dataset provides a rendered synthetic depth frame corresponding to each of the real images. While we sample the real images only from the frontal camera, which is used for the standard training set, for our synthetic data set we follow Rad et al. (2018b) and use images, which have been rendered from the viewpoints of each of the three cameras. That is, we use all 218,271 synthetic samples provided with the dataset. Note, for the distribution matching loss, we sample real and synthetic data only from the 72,757 samples from the frontal view.

In Chapter 4 we performed the view prediction experiments on a subset of the NYU dataset as the camera setup was changed during capturing the dataset but the introduced method assumes a fixed setup. Nevertheless, after the setup change one of the camera viewpoints roughly corresponded to another camera pose from before the change of the setup. Since the method we want to test here is not solely based on the view prediction task, we use the data from the roughly corresponding camera view after the setup change in order to exploit all 72,757 real samples in the dataset and hence more fairly compare to other works.

Data augmentation We used online data augmentation. That is, each time we sample a specific image we also sample new transformation parameters. In the experiments for this chapter we randomly rotate the loaded image, randomly sample the location of the crop and add white noise to the depth values. The rotation angle is uniformly sampled from $[-60°, 60°]$

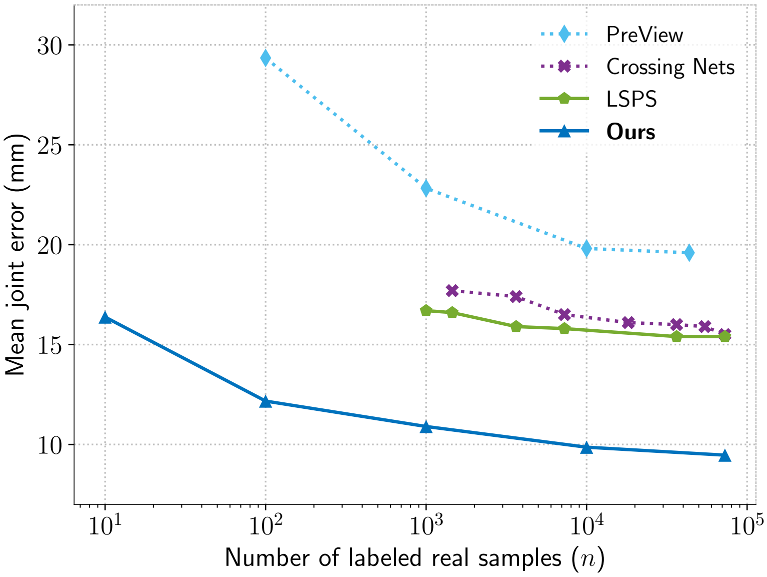


Figure 5.3: **Comparison to semi-supervised approaches.** Comparison to the recent approaches *PreView* (Poier et al., 2018), *Crossing Nets* (Wan et al., 2017) and *LSPS* (Abdi et al., 2018) for different numbers of labeled real samples n.

and the location offset as well as white noise is sampled from a normal distribution with $\sigma = 5\,\text{mm}$.

5.3.2 Comparison to state-of-the-art approaches

We start with a quantitative comparison to state-of-the-art approaches on hand pose estimation. To this end, we compare to the results of recent semi-supervised approaches, but also show how our method performs in comparison to fully supervised approaches when training on the full dataset.

Comparison to semi-supervised methods Only a few approaches have recently targeted the semi-supervised setting for hand pose estimation: we compare to *Crossing Nets* (Wan et al., 2017), *LSPS* (Abdi et al., 2018) and our method introduced in the previous chapter (*PreView*). Figure 5.3 shows the results for different numbers of labeled real samples. Note, that only *LSPS* (Abdi et al., 2018) exploits synthetic and real data jointly, tackling the domain gap. We compare to the results published by the authors, which are provided for different numbers of labeled samples. Nevertheless, we can see that our method outperforms their results independent of the number of labeled real samples.

Table 5.1: **Comparison to state-of-the-art.** Mean joint error (ME) for training with all labeled real samples from the NYU dataset for recent state-of-the-art approaches, baselines and our method.

Method	ME (mm)
DISCO Nets (Bouchacourt et al., 2016) (NIPS 2016)	20.7
Crossing Nets (Wan et al., 2017) (CVPR 2017)	15.5
LSPS (Abdi et al., 2018) (BMVC 2018)	15.4
Weak supervision (Neverova et al., 2017) (CVIU 2017)	14.8
Lie-X (Xu et al., 2017) (IJCV 2017)	14.5
3DCNN (Ge et al., 2017) (CVPR 2017)	14.1
REN-9x6x6 (Wang et al., 2018) (JVCI 2018)	12.7
DeepPrior++ (Oberweger and Lepetit, 2017) (ICCVw 2017)	12.3
Pose Guided REN (Chen et al., 2018a) (Neurocomputing 2018)	11.8
SHPR-Net (Chen et al., 2018b) (IEEE Access 2018)	10.8
Hand PointNet (Ge et al., 2018) (CVPR 2018)	10.5
Dense 3D regression (Wan et al., 2018) (CVPR 2018)	10.2
V2V single model (Moon et al., 2018) (CVPR 2018)	9.2
V2V ensemble (Moon et al., 2018) (CVPR 2018)	8.4
Feature mapping (Rad et al., 2018b) (CVPR 2018)	7.4
Synthetic only	21.3
Real only	14.7
Real and Synthetic	13.1
Ours	9.5

Comparison on full dataset We compare to fully supervised state-of-the-art approaches when employing all labeled data. We want to stress that our work does not focus on the case where a huge number of labeled real samples, roughly covering the space of poses in the test set, is readily available. We show this comparison, rather, to prove the competitiveness of our implementation. Table 5.1 shows the comparison including some of our baselines. We can see that the results of our system are within the top state-of-the-art approaches. Comparing the results for state-of-the-art approaches in Table 5.1 with the results of our method when using a smaller number of labeled real samples in Figure 5.3, we see that our method performs similar to recent state-of-the-art approaches even using only a small fraction of the labeled real samples. Also note that several of the most recent methods focus on improved input and/or output representations (Chen et al., 2018b; Ge et al., 2018; Moon et al., 2018; Wan et al., 2018), which are orthogonal to our work.

The comparisons in this section are based upon the numbers published by the authors. That is, these comparisons disregard differences in the used data subsamples, models, architectures, and other specificities. For a better evaluation of the contribution of this chapter we investigate the different ingredients of our method based on the same experimental setup in the next section.

5.3.3 Ablation study

In the ablation study we aim to compare our method to baselines based on the same experimental setup and investigate how effective our contributions are. To this end, we use the same architecture and train it with different data: only with labeled real data, only with synthetic data, with labeled real and synthetic, or with labeled real, synthetic and additional unlabeled real data. Figure 5.4 shows the results for different numbers of labeled real samples. We compare to different variants of our method denoted *Real+Synth. | **, where the asterisk (*) acts as a placeholder for how we train the mapping and thus exploit unlabeled data. That is, we compare the full implementation of our method (*Real&Synth. | Full*, Eq. (5.1)) and two ablated variants: One variant where the exploitation of unlabeled data is only based on the adversarial loss term (*Real+Synth. | Distr. Match*), and another variant where only the view prediction objective is used for unlabeled data (*Real+Synth. | View Pred.*).

We see that each of the individual loss terms yields a significant performance gain compared to the baseline system, which uses real and synthetic data but cannot exploit unlabeled data. The additional gain of the full system over the variants with only one of the loss terms is more enhanced for a small number of labeled real samples n and only small for large n, but consistent over all n.

In Figure 5.5 we compare the results of training with and without unlabeled data qualitatively. Before investigating the error cases, here, we especially focus on the question for which samples our method actually improved the results. We often find samples which are significantly distorted, *i.e.*, the depth map contains holes, parts of the hand or even complete fingers are missing.

5.3.4 Error case analysis

We aim to investigate for which samples our full method makes the largest errors. We analyze the error cases for our model trained with 100 labeled real samples. Representative samples from the frames with largest mean error are shown in Figure 5.6, samples from the frames with largest maximum

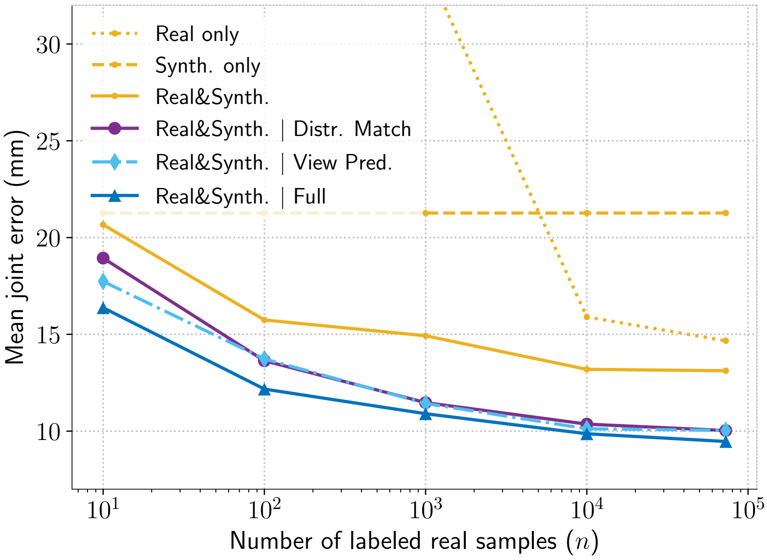


Figure 5.4: **Ablation experiments.** How different aspects of our method influence the performance over different numbers of labeled real samples n. *Real* and *Synth.* specifies whether real or synthetic data was used and further descriptions identify different variants of our method. See text for details.

error are shown in Figure 5.7. Note that many of the frames with large error are essentially near duplicates of the the shown error cases since the test set is actually a continuous sequence with similar neighboring frames. We find that our model has problems especially if none of the fingers is clearly visible in the depth frame, *i.e.*, the frame has a "blob-like" appearance.

For the frames for which our model had the largest problems with, we additionally search for the nearest neighbors in the training set. We find the nearest neighbors based on the average joint distance between the corresponding ground truth annotations (after shifting the annotations to the origin to ignore translations). Figure 5.8 shows the nearest neighbors for some selected test samples. We find that for some samples there are no close nearest neighbors in the training set, and we hypothesize that for such "blob-like" structures it is especially difficult to obtain valuable feedback from the view prediction objective. Also note, that the model we are analyzing was trained on only 100 labeled real samples and the labels for the nearest neighbors shown in Figure 5.8 were not used.

5.3.5 Latent space analysis

In a final set of experiments we investigate the learned latent representation. We are especially interested in how the introduced method affects the shared latent space of real and synthetic data.

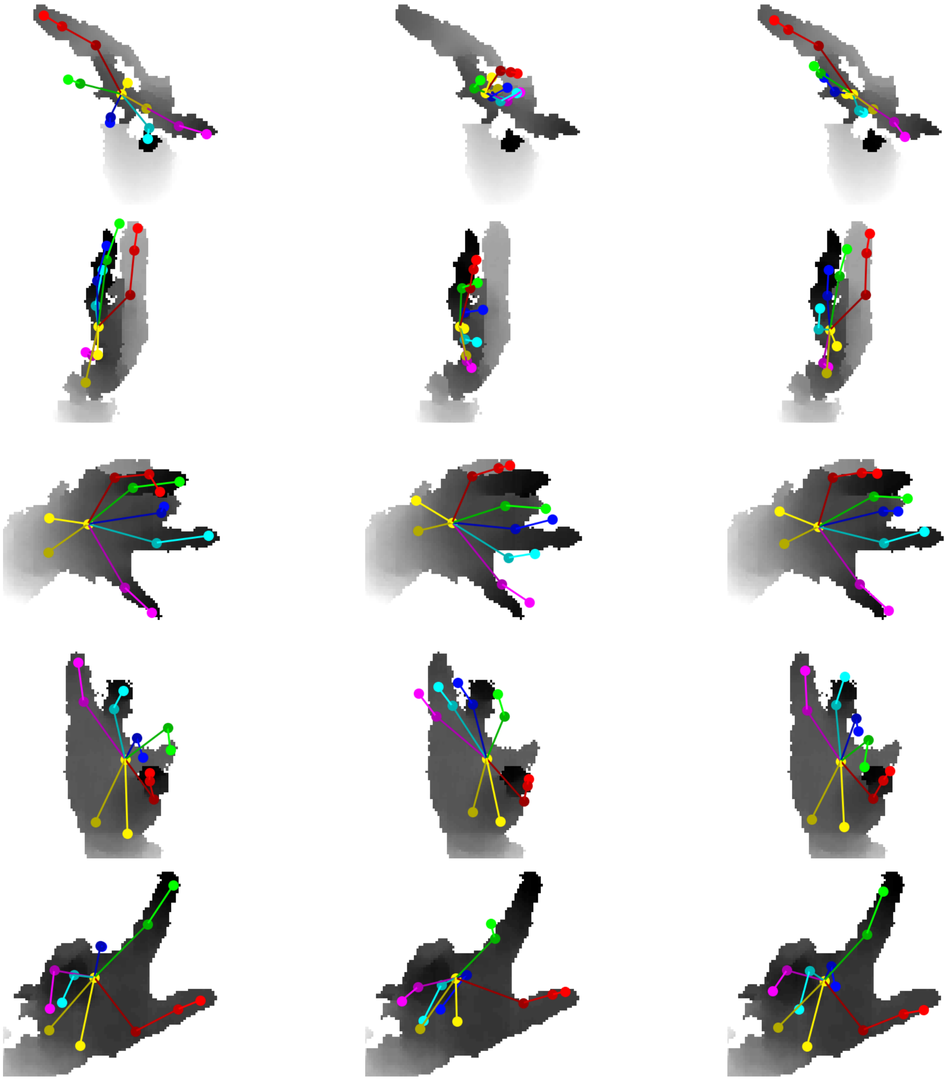


Figure 5.5: **Qualitative results.** Left: ground truth. Middle: baseline trained with labeled data (synthetic and 100 real). Right: our result from training with the same labeled samples and additional unlabeled real data. We find that our method improves results especially for highly distorted images and difficult poses. Best viewed in color.

Visualization We compute the latent representations of corresponding real and synthetic samples from the validation set and visualize the representations using t-SNE (van der Maaten and Hinton, 2008). From the t-SNE visualization in Figure 5.9 we can see that the real and synthetic data is well aligned and that the aligned data points correspond to similar poses. This is illustrated by the depth images for exemplary parts of the representation. Nevertheless, such visualizations have to be interpreted with caution (*c.f.*,

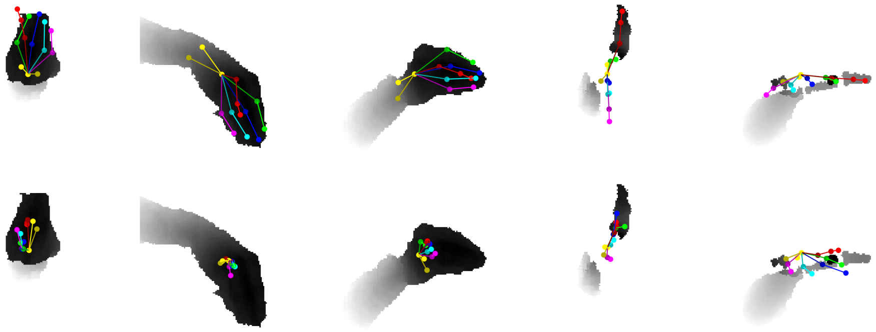


Figure 5.6: **Frames with largest mean error.** Test samples overlaid with ground truth (top row) and the predictions of our model (bottom row). Note, 90 of the 100 frames with the largest mean error are variations of the leftmost three frames.

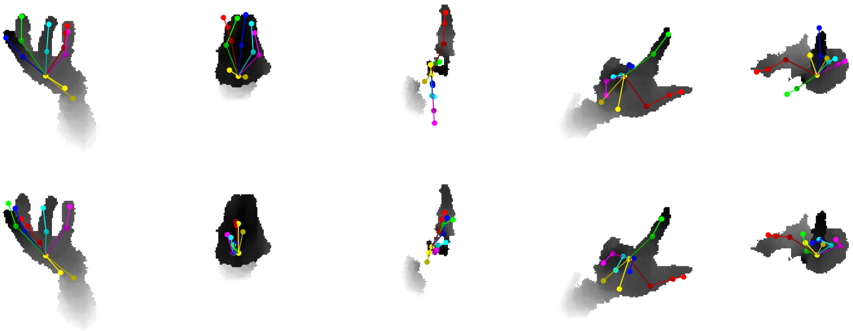


Figure 5.7: **Frames with largest maximum error.** Test samples overlaid with ground truth (top row) and the predictions of our model (bottom row). The errors are mainly due to strongly distorted samples and annotation errors (*e.g.*, left most sample). Note, 79 of the 100 frames with largest maximum error are variations of the three leftmost frames.

e.g., (Wattenberg et al., 2016)). Hence, in the following, we try to get more insights from analyzing the distances directly.

Distance distributions To better investigate how our contributions affect the latent space distributions, we again make use of the fact that we have corresponding real and synthetic validation samples. We compute the distances of the latent representations of corresponding real and synthetic samples and compare the distribution of these distances for different experiments. In Figure 5.10 we compare the distance distribution for: (i) a baseline experiment which was trained jointly with synthetic and 100 labeled real samples, and (ii) our approach, which was trained with the same labeled data but additionally employs unlabeled data. We can see, that despite the weak supervision from the unlabeled data, *i.e.* correspondence is

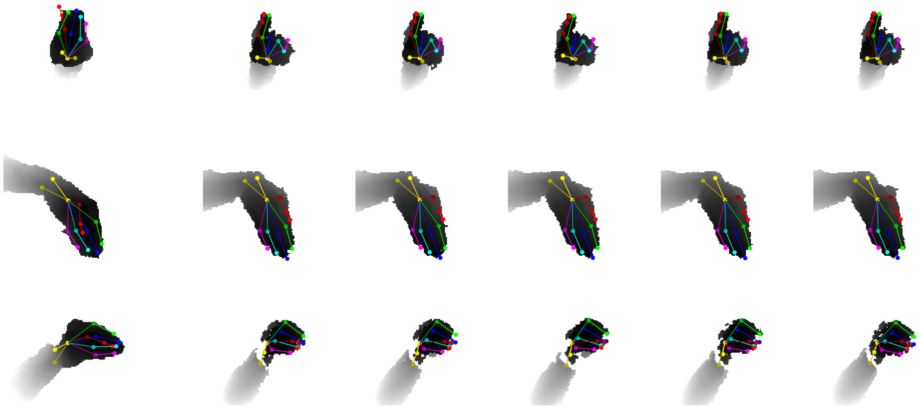


Figure 5.8: **Nearest neighbors in training set.** The test samples with largest error (*c.f.*, Figure 5.6 and Figure 5.7) and their (pose based) nearest neighbors in the training set. Leftmost column shows the test sample, the remaining columns show the corresponding nearest neighbors from the training set. Note, the training samples were used unlabeled only.

not known and only the additional loss terms described in Sec. 5.2.3 can be used to match the data, the distance between the corresponding validation samples are clearly smaller.

Example view predictions Finally, we compare examples for view predictions of our method given input from either real or synthetic data. This is interesting, since a possible drawback of the view prediction objective is that the generator g might try to discriminate between real and synthetic data in order to predict the appearance accurately, as has been discussed in Sec. 5.2.3. However, by looking at the predicted views (*c.f.*, Figure 5.11) we see that this is not the case. We rather find that the predictions are nearly equivalent for real and synthetic samples with the same pose, again indicating that similar poses are close together in the latent space – independent of the domain – which was the intention of the contributions introduced in this chapter.

5.4 Conclusions

In this chapter we focused on the exploitation of synthetic data for the task of 3D hand pose estimation from depth images. Most importantly, we showed that the existing domain gap between real and synthetic data, which hampers the exploitation, can be reduced using mainly unlabeled real data. To this end, we introduced two auxiliary objectives, which ensured that input images exhibiting similar poses are close together in a shared

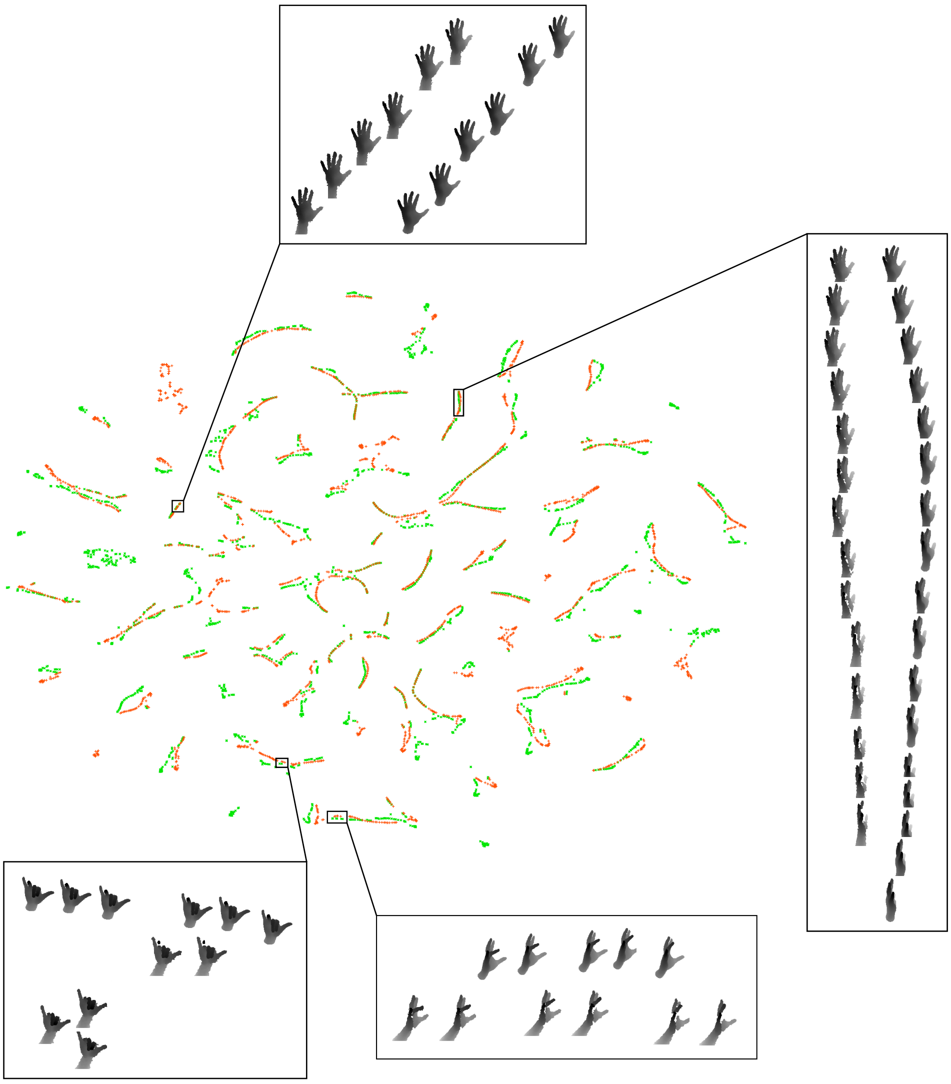


Figure 5.9: **Visualization of latent representations.** t-SNE visualization of the learned latent representation of real (green; ✖) and synthetic (orange; ✚) samples from the validation set. Simultaneously, real and synthetic samples as well as similar poses are aligned in the latent representation, while only 100 corresponding real and synthetic images are employed during training. Note, if necessary, we moved the visualized depth images slightly apart, so, that they do not overlap. Best viewed in color with zoom.

latent space – independent of the domain they are from. We showed that our method outperforms many recent state-of-the-art approaches using a surprisingly small fraction of the labeled real samples they use.

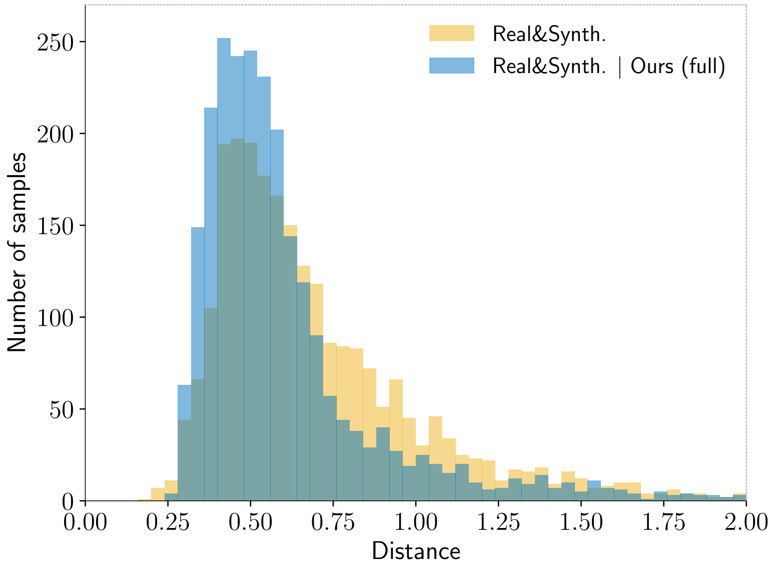


Figure 5.10: **Distributions of latent space distances between corresponding real and synthetic samples.** Comparison of the distance distributions for our method (blue) and a baseline (yellow). The higher peak for lower distances shows that our method moves corresponding real and synthetic data closer together. See text for details.

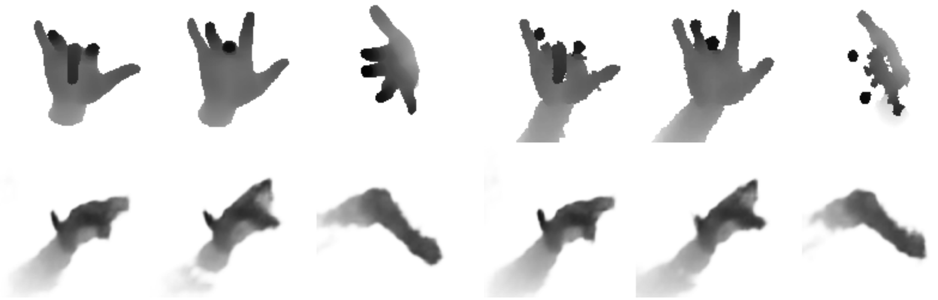


Figure 5.11: **Example view predictions for real and synthetic input. Top:** Three corresponding synthetic (left) and real (right) validation images. **Bottom:** Predicted views for synthetic (left) and real (right) input.

CHAPTER 6

Conclusion

Contents

In this thesis we introduced several distinct methods to learn to estimate the hand pose – all of which yield competitive results despite requiring only a subset of the labels, implying a significantly reduced labeling effort. After providing the relevant background for such an endeavor, we started by pointing out a way to combine a learned model with prior knowledge about the target object in an analysis-by-synthesis manner. Subsequently, we introduced a method to exploit unlabeled data by requiring the model to learn about the relations between different views of the hand. Finally, we showed how synthetic data can be better exploited by mitigating the domain gap.

In the first of our technical contributions we introduced a way to exploit prior knowledge about the hand by employing a 3D hand model. We devised a hybrid approach, combining a data-driven with a 3D hand model-based part. In this method we fit the hand model to the output of the data-driven part. Naturally, by fitting the hand model, the system can always provide an anatomically valid solution. Additionally, we designed the data-driven part to output a distribution over possible locations for each joint. We showed that the model-based part can further improve the results by considering this distribution and thus implicitly considering for which estimates the data-driven part is certain and for which it is uncertain. While a smaller training dataset immediately leads to an increased error of the learned data-driven approach, we showed that by employing the model-based optimization in this way we are able to partly overcome this issue.

The combination of a data-driven part with a model-based part yields improved results but is still significantly impaired when the training set for the data-driven part becomes substantially smaller since the model-based optimization is not able to fix gross errors. Such errors usually occur for samples, which are too far from the distribution of the training set. That is, while the labeling effort can be reduced with this method, still a reasonably large amount of labeled training data is required for many applications.

With the goal to have the model learn from a large training set without increasing the labeling effort we developed a method to exploit unlabeled data. We showed that by learning to predict how the hand would look from a different view and thus learn about the relations of the appearance of the same pose in different viewpoints the model learns to extract information, which is closely related to the pose of the hand. Being able to learn such a pose specific representation from unlabeled data, we also pointed out a way to use the unlabeled data in combination with some labeled data. Employing labeled and unlabeled data together we showed that we can achieve similar accuracy when training with one order of magnitude less labeled data.

Finally, to obtain more supervision for the still complex mapping from a pose specific latent representation to the desired targets (*i.e.*, joint positions), we exploited labeled data from a different domain; namely synthetic data. Employing synthetic data provides a possibility to circumvent the requirement for large datasets of labeled real data. Nevertheless, models trained solely on synthetic data show a significant drop in performance compared to models trained on real data. This is due to the domain gap, *i.e.*, the difference between the distribution of the synthetic training and the real test data. To mitigate the domain gap without requiring a large amount of labeled real data, we enforced that the feature distributions of real and synthetic data are aligned and simultaneously ensured that similar poses are mapped to similar locations in the latent feature space. We showed that using our method we can reach a similar performance to the baseline without our contributions using only about 10 to 100 labeled real samples compared to more than 70,000 for the baseline.

6.1 Discussion

We have set an ambitious goal for this thesis: to develop a method, which is able to learn to estimate the pose of the human hand without requiring manual supervision. There are obvious flaws in this formulation of our quest. To state whether a system has learned a task or not, we naturally need to define a criterion based on which we can make such a decision. It is, however, not straightforward to define such a criterion. We could even argue that in many cases it might be impossible to come up with a universal criterion to decide whether a system has solved a task or not. This is because such a criterion always depends on the actual application. For example, the current state-of-the-art approaches for hand pose estimation might be readily sufficient for applications where the range of poses is known a priori, the occurring occlusions are limited, other objects in the scene do not pose any difficulties for segregating the hand and computational power is not a real issue. Hence, while the task can be considered solved for a range of applications it is rather easy to imagine conditions under which they would fail; *e.g.*, computational constraints can be virtually arbitrarily aggravated. Hence, to judge our contributions we followed a more practical way, which is still generically applicable to any specific application. We studied how far we can reduce the manual labeling effort without sacrificing accuracy.

We showed that using a combination of methods we can achieve results close to the current state-of-the-art with only a fraction of the labeled real samples. Moreover, in Chapter 5 we were able to show that with the proposed adaptations of the baseline it is possible to even improve the results, while discarding 99.9% of the labeled real data and using unlabeled data instead.

Nevertheless, despite the improved results, the introduced methods exhibit limitations. In general, all the introduced methods still require some number of labeled real samples in order to perform well. While, for the method introduced in Chapter 5, this is a comparably low number, the labeled real samples still enable a significant improvement of the results and are thus crucial for its success. Furthermore, the intuition that we can exploit synthetic data by enforcing the latent distributions of real and synthetic data to be aligned, is inherently based on the assumption that the pose distribution of real and synthetic data is similar. A requirement, which has to be considered when capturing the real data or generating synthetic data. For the hybrid approach we introduced in Chapter 3 we assumed to know the size of the hand in advance and used a model with fixed distances between the joint positions. Hence, the model has to be specifically adapted to each actor. This is a prominent problem for related approaches and has been tackled in various ways. A common example is to require a calibration phase for each user in advance (Tan et al., 2016; Taylor et al., 2014), but also approaches to adapt the model during test time have been proposed (Makris and Argyros, 2015; Tkach et al., 2017). Moreover, to compare to related work we focused our experiments on existing datasets, which provide labels for all samples. That is, to study the effect when only a fraction of the training data is labeled we used the labels only for a subsample of the full set and used all remaining samples without labels. In practice, however, one would of course use all labeled samples which are available and capture additional unlabeled samples using the target setup to better cover the pose space. To this end, it is still unclear how the methods would behave if more labeled and unlabeled data would be provided. That is, it would be interesting how effective the methods would be in cases where there is already a large number of labeled data and additionally several orders of magnitude more unlabeled data can be exploited. Note, while studies showed the effectiveness of evermore data for other tasks – even if it has noisy labels or is unlabeled (Goyal et al., 2019; Sun et al., 2017a) – this question is open to be investigated for the methods we proposed in this thesis.

6.2 Future directions

This thesis clearly failed to achieve its ultimate goal to develop a method which is able to learn systems without any manual labeling effort for a broad range of applications. Still, there is clearly no need to worry that no further improvements are possible, which can bring us closer to the ambitious goal.

One option is to incorporate temporal information. For our work we excluded temporal information on purpose. The system should not require to have temporal information available. It should rather be able to estimate a pose given a single frame. In this way the system can be straightforwardly employed for initialization at the beginning of a sequence and to re-initialize after, *e.g.*, the hand was outside the camera view. However, temporal information naturally provides a cue for cases, which are otherwise difficult to solve at all. For example, at test time short temporary occlusions during movement can often be reasonably resolved when employing information from neighboring frames in time. Furthermore, being able to estimate the pose from a single frame does not necessarily preclude the system from being able to exploit temporal information as soon as it is available.

Such temporal information can also be exploited during training time. For example, again following the idea of self-supervised learning, a model can be trained from unlabeled data to associate corresponding parts of the hand over neighboring frames in time (*c.f.* Section 2.3.3). Employing such ideas, we might be able to learn a more robust model without increasing the labeling effort.

Another option is to incorporate a 3D hand model in the training phase. The 3D hand model provides an opportunity to obtain feedback for training a data-driven model. The realization of such an approach can be similar to the way a model-based step can refine the results of a data-driven part at test time; but in this case it would be used to obtain feedback for training the data-driven part. Initial works following such an approach have been published recently (*c.f*. Section 2.3.5 and 2.5.3).

Finally we want to mention that the discussed ideas could also be adapted to work with color images. While working solely with color inputs poses different challenges (*e.g.*, additional variations due to lighting or skin color, foreground segmentation is usually a larger issue than with depth images, and depth information eases accurate 3D estimates), it also bears several advantages. For example, when the hand is interacting with objects it can be easier to segregate the hand and objects as they often exhibit a different color. Additionally, the cameras are usually significantly cheaper and consume less energy making them also more applicable for mobile devices.

However, obtaining accurate 3D pose labels for arbitrary color images is usually an even larger issue than for depth images. The difficulty is further aggravated if the hand is interacting with objects or other hands, as the factors of variation are again significantly increased in this case. Hence, the ideas to reduce the labeling effort, which we introduced in this thesis, are of large interest for this direction.

In line with this, we believe that pose estimation systems which work with a largely reduced labeling effort, will be much easier applicable to novel tasks. Reducing the effort to the effort which is needed to capture data will make such systems much more accessible. Researchers and engineers which are unable to invest in extensive labeling would still be able to foster technological progress.

APPENDIX A

List of publications

Contents

For the sake of completeness of this thesis this section lists the publications I authored and coauthored.

A.1 Publications underlying the thesis

The thesis is based upon the following three peer-reviewed publications.

Hybrid One-Shot 3D Hand Pose Estimation by Exploiting Uncertainties

Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof and Antonis A. Argyros
In *Proceedings of the British Machine Vision Conference (BMVC)*
September 2015, Swansea, United Kingdom
(Accepted for oral presentation)

Learning Pose Specific Representations by Predicting Different Views

Georg Poier, David Schinagl and Horst Bischof
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR)
June 2018, Salt Lake City, United States of America
(Accepted for spotlight presentation)

MURAUER: Mapping Unlabeled Real Data for Label AUsteRity

Georg Poier, Michael Opitz, David Schinagl and Horst Bischof
In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*
January 2019, Waikoloa Village, United States of America
(Accepted for spotlight presentation)

A.2 Further co-/authored publications

I authored and coauthored a number of additional publications, which are listed here for the sake of completeness.

Multi-Cue Learning and Visualization of Unusual Events

René Schuster, Samuel Schulter, Georg Poier, Martin Hirzer, Josef A. Birchbauer, Peter M. Roth, Horst Bischof, Martin Winter and Peter Schallauer
In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop on Visual Surveillance*
November 2011, Barcelona, Spain
(Accepted for poster presentation)

OUTLIER – Online Learning and Visualization of Unusual Events

Josef A. Birchbauer, Samuel Schulter, René Schuster, Georg Poier, Martin Winter, Peter Schallauer, Peter M. Roth and Horst Bischof
In *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS) Demo Session*
November 2011, Klagenfurt, Austria
(Extended abstract)

Text Localization in Unconstrained Images

Georg Poier, Jürgen Hatzl, Stefan Kluckner, Peter M. Roth and Horst Bischof
In *Proceedings of the Computer Vision Winter Workshop (CVWW)*

February 2012, Mala Nedelja, Slovenia
(Accepted for poster presentation)

Hough Forests Revisited: An Approach to Multiple Instance Tracking from Multiple Cameras

Georg Poier, Samuel Schulter, Sabine Sternig, Peter M. Roth and Horst Bischof
In *Proceedings of the German Conference on Pattern Recognition (GCPR)*
September 2014, Münster, Germany
(Accepted for poster presentation)

Navigation Assistance and Guidance of Older Adults across Complex Public Spaces: the DALi Approach

Luigi Palopoli, Antonis A. Argyros, Josef A. Birchbauer, Alessio Colombo, Daniele Fontanelli, Axel Legay, Andrea Garulli, Antonello Giannitrapani, David Macii, Federico Moro, Payam Nazemzadeh, Pashalis Padeleris, Roberto Passerone, Georg Poier, Domenico Prattichizzo, Tizar Rizano, Luca Rizzon, Stefano Scheggi and Sean Sedwards
Intelligent Service Robotics 8(2):77-92
March 2015

Interactive Segmentation of Rock-Art in High-Resolution 3D Reconstructions

Matthias Zeppelzauer, Georg Poier, Markus Seidl, Christian Reinbacher, Christian Breiteneder, Horst Bischof and Samuel Schulter
In *Proceedings of Digital Heritage*
September/October 2015, Granada, Spain
(Accepted for oral presentation; Winner of the best paper award)

Interactive 3D Segmentation of Rock-Art by Enhanced Depth Maps and Gradient Preserving Regularization

Matthias Zeppelzauer, Georg Poier, Markus Seidl, Christian Reinbacher, Samuel Schulter, Christian Breiteneder and Horst Bischof
ACM Journal on Computing and Cultural Heritage (JOCCH)
December 2016

The 3D-PITOTI Project with a Focus on Multi-Scale 3D Reconstruction using Semi-Autonomous UAVs

Christian Mostegel, Georg Poier, Christian Reinbacher, Manuel Hofer, Friedrich Fraundorfer, Horst Bischof, Thomas Höll, Gert Holler and Axel Pinz
In *Proceedings of the Austrian Association of Pattern Recognition (AAPR) & Austrian Robotic Workshop (ARW) Joint Workshop*
May 2016, Wels, Austria
(Extended abstract; Accepted for oral presentation)

Grid Loss: Detecting Occluded Faces

Michael Opitz, Georg Waltner, Georg Poier, Horst Possegger and Horst Bischof
In *Proceedings of the European Conference on Computer Vision (ECCV)*
October 2016, Amsterdam, Netherlands
(Accepted for poster presentation)

Loss-Specific Training of Random Forests for Super-Resolution

Alexander Grabner, Georg Poier, Michael Opitz, Samuel Schulter and Peter M. Roth
In *Proceedings of the Computer Vision Winter Workshop (CVWW)*
February 2017, Retz, Austria
(Accepted for oral presentation)

The 3D-Pitoti Dataset: A Dataset for high-resolution 3D Surface Segmentation

Georg Poier, Markus Seidl, Matthias Zeppelzauer, Christian Reinbacher, Martin Schaich, Giovanna Bellandi, Alberto Marretta and Horst Bischof
In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*
June 2017, Florence, Italy
(Accepted for oral presentation)

Being Lazy at Labelling for Pose Estimation

Georg Poier, David Schinagl and Horst Bischof
In *Proceedings of the Austrian Association of Pattern Recognition (AAPR)*

May 2018, Hall/Tyrol, Austria
(Extended abstract; Accepted for oral presentation)

APPENDIX B

List of acronyms

AUC	area under the curve
CCA	Canonical Correlation Analysis
CNN	Convolutional Neural Network
CoM	center of mass
DoF	degree of freedom
DPM	Deformable Part Model
FS	frame-based success rate
GAN	Generative Adversarial Network
JS	joint-based success rate
LSTM	Long Short-Term Memory
ME	mean joint error
NCE	Noise Contrastive Estimation
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RF	Random Forest
ToF	Time-of-Flight
VAE	Variational Autoencoder

Bibliography

Abdi, M., Abbasnejad, E., Lim, C. P., and Nahavandi, S. (2018). 3d hand pose estimation using simulation and partial-supervision with a shared latent space. In *Proc. British Machine Vision Conf.* 93, 94, 101, 104, 105

Albrecht, I., Haber, J., and Seidel, H.-P. (2003). Construction and animation of anatomically based human hand models. In *Proc. Eurographics Symposium on Computer Animation*. 34, 47, 72, 81

Amit, Y. and Geman, D. (1994). Randomized inquiries about shape; an application to handwritten digit recognition. Technical Report 401, Department of Statistics, University of Chicago, IL. 45

Andrew, G., Arora, R., Bilmes, J. A., and Livescu, K. (2013). Deep Canonical Correlation Analysis. In *Proc. Int'l Conf. on Machine Learning*. 22

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *Proc. Int'l Conf. on Learning Representations*. 30

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *ArXiv e-prints*, abs/1701.07875. 101

Baak, A., Müller, M., Bharaj, G., Seidel, H.-P., and Theobalt, C. (2011). A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. IEEE Int'l Conf. on Computer Vision*. 42

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 25

Ballan, L., Taneja, A., Gall, J., Van Gool, L., and Pollefeys, M. (2012). Motion capture of hands in action using discriminative salient points. In *Proc. European Conf. on Computer Vision*. 35, 38, 44

Becker, S. and Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163. 22

Berthelot, D., Carlini, N., Goodfellow, I. J., Papernot, N., Oliver, A., and Raffel, C. (2019). Mixmatch: A holistic approach to semi-supervised learning. *ArXiv e-prints*, abs/1905.02249. 13

Blum, A. and Mitchell, T. M. (1998). Combining labeled and unlabeled data with co-training. In *Proc. Conf. on Computational Learning Theory*. 22

Bouchacourt, D., Mudigonda, P. K., and Nowozin, S. (2016). DISCO nets : Dissimilarity coefficients networks. In *Proc. Neural Information Processing Systems*. 105

Boukhayma, A., Bem, R. d., and Torr, P. H. (2019). 3D hand shape and pose from images in the wild. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 25, 38, 39

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. 45

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a siamese time delay neural network. In *Proc. Neural Information Processing Systems*. 28

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75. 28

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA. 11, 12, 13

Chen, C.-H., Tyagi, A., Agrawal, A., Drover, D., MV, R., Stojanov, S., and Rehg, J. M. (2019a). Unsupervised 3d pose estimation with geometric self-supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 24

Chen, X., Lin, K., Liu, W., Qian, C., Wang, X., and Lin, L. (2019b). Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 24

Chen, X., Wang, G., Guo, H., and Zhang, C. (2018a). Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*. 105

Chen, X., Wang, G., Zhang, C., Kim, T.-K., and Ji, X. (2018b). SHPR-Net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6:43425–43439. 105

Chu, C., Zhmoginov, A., and Sandler, M. (2017). CycleGAN, a master of steganography. *ArXiv e-prints*, abs/1712.02950. 30

Coates, A., Ng, A. Y., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proc. Int'l Conf. on Artificial Intelligence and Statistics*. 72, 77

Colman, A. M. (2015). *A Dictionary of Psychology*. Oxford University Press. 3

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619. 12, 46

Criminisi, A., Shotton, J., and Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227. 45

Csurka, G., editor (2017). *Domain Adaptation in Computer Vision Applications*. Advances in Computer Vision and Pattern Recognition. Springer. 26, 28

Dai, W., Yang, Q., Xue, G., and Yu, Y. (2008). Self-taught clustering. In *Proc. Int'l Conf. on Machine Learning*. 29

Dantone, M., Gall, J., Leistner, C., and Van Gool, L. (2013). Human pose estimation using body parts dependent joint regressors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 44

Dantone, M., Gall, J., Leistner, C., and Van Gool, L. (2014). Body parts dependent joint regressors for human pose estimation in still images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(11):2131–2143. 43, 44

de La Gorce, M., Fleet, D. J., and Paragios, N. (2011). Model-based 3d hand pose estimation from monocular video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805. 35, 36, 37, 42, 67

de Sa, V. R. (1993). Learning classification with unlabeled data. In *Proc. Neural Information Processing Systems*. 13, 14, 21

Decker, J. H., Lourenco, F. S., Doll, B. B., and Hartley, C. A. (2015). Experiential reward learning outweighs instruction prior to adulthood. *Cognitive, Affective, & Behavioral Neuroscience*, 15(2):310–320. 10

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *ArXiv e-prints*, abs/1810.04805. 15

Dibra, E., Wolf, T., Öztireli, A. C., and Gross, M. H. (2017). How to refine 3d hand pose estimation from unlabelled depth data? In *Proc. IEEE Int'l Conf. on 3D Vision*. 24, 38, 39

Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proc. IEEE Int'l Conf. on Computer Vision*. 10, 11, 14, 16, 17, 18, 21, 68, 79

Doersch, C. and Zisserman, A. (2017). Multi-task self-supervised visual learning. In *Proc. IEEE Int'l Conf. on Computer Vision*. 18

Doersch, C. and Zisserman, A. (2019). Sim2real transfer learning for 3d pose estimation: motion to the rescue. *ArXiv e-prints*, abs/1907.02499. 30

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. Int'l Conf. on Machine Learning*. 28, 31

Donner, R., Menze, B. H., Bischof, H., and Langs, G. (2013). Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. *Medical Image Analysis*, 17(8):1304–1314. 34

Dosovitskiy, A. and Koltun, V. (2017). Learning to act by predicting the future. In *Proc. Int'l Conf. on Learning Representations*. 10, 13, 14, 21

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M. A., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In *Proc. Neural Information Processing Systems*. 11, 17, 72, 77

Douvantzis, P., Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2013). Dimensionality reduction for efficient single frame hand pose estimation. In *Proc. Int'l Conf. on Computer Vision Systems*. 34

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T., and Efros, A. A. (2018). Investigating human priors for playing video games. In *Proc. Int'l Conf. on Machine Learning*. 32

Dyer, C. (2014). Notes on noise contrastive estimation and negative sampling. *ArXiv e-prints*, abs/1410.8251. 15

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*, 108(1-2):52–73. 2, 3, 33, 34

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645. 42, 43

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *Int'l Journal of Computer Vision*, 61(1):55–79. 42

Fernando, B., Bilen, H., Gavves, E., and Gould, S. (2017). Self-supervised video representation learning with odd-one-out networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 19, 20

Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 22(1):67–92. 32, 42

Fukushima, K. (1980). Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202. 75

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *Proc. IEEE Int'l Conf. on Computer Vision*. 42

Ganapathi, V., Plagemann, C., Koller, D., and Thrun, S. (2010). Real time motion capture using a single time-of-flight camera. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 4, 38

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35. 28, 31

Garg, R., Kumar, B. G. V., Carneiro, G., and Reid, I. D. (2016). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proc. European Conf. on Computer Vision*. 11, 23, 69

Ge, L., Cai, Y., Weng, J., and Yuan, J. (2018). Hand PointNet: 3d hand pose estimation using point sets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 103, 105

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 105

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741. 42

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *Proc. Int'l Conf. on Learning Representations*. 10, 18

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 28

Girshick, R. B., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. W. (2011). Efficient regression of general-activity human poses from depth images. In *Proc. IEEE Int'l Conf. on Computer Vision*. 43

Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 23, 69

Gomez-Bigorda, L., Patel, Y., Rusiñol, M., Karatzas, D., and Jawahar, C. V. (2017). Self-supervised learning of visual features through embedding images into text topic spaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 14, 21

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. Neural Information Processing Systems*. 74, 93, 100

Gordo, A. and Larlus, D. (2017). Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 21

Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch SGD: training imagenet in 1 hour. *ArXiv e-prints*, abs/1706.02677. 102

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. (2019). Scaling and benchmarking self-supervised visual representation learning. *ArXiv e-prints*, abs/1905.01235. 11, 18, 116

Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In *Proc. Neural Information Processing Systems*. 12

Guo, H., Wang, G., Chen, X., and Zhang, C. (2017). Towards good practices for deep 3d hand pose estimation. *ArXiv e-prints*, abs/1707.07248. 65, 67

Gutmann, M. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361. 15

Hamer, H., Schindler, K., Koller-Meier, E., and Van Gool, L. (2009). Tracking a hand manipulating an object. In *Proc. IEEE Int'l Conf. on Computer Vision*. 43

Hartley, C. A. and Somerville, L. H. (2015). The neuroscience of adolescent decision-making. *Current Opinion in Behavioral Science.*, 5:108–115. 10

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, New York, NY. 11, 12

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 102

He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. (2004). Multiscale conditional random fields for image labeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 42

Held, R. and Hein, A. (1963). Movement-produced stimulation in the development of visually guided behavior. *Journal of Comparative and Physiological Psychology*, 56(5):872–876. 10

Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). Transforming autoencoders. In *Proc. Int'l Conf. on Artificial Neural Networks*. 16, 68

Hinton, G. E. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. 15, 68

Hinton, G. E., Sejnowski, T. J., and Poggio, T. A. (1999). *Unsupervised Learning: Foundations of Neural Computation*. MIT Press, Cambridge, MA. 12

Hollister, A., Buford, W. L., Myers, L. M., Giurintano, D. J., and Novick, A. (1992). The axes of rotation of the thumb carpometacarpal joint. *Journal of Orthopaedic Research*, 10:454–460. 34

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441,498–520. 12, 16, 79

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3-4):321–377. 21

Huang, H., Huang, Q., and Krähenbühl, P. (2018). Domain transfer through deep activation matching. In *Proc. European Conf. on Computer Vision*. 10, 31

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101. 73

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2016). Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Trans. on Graphics*, 35(4):110:1–110:11. 17, 20

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 21

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int'l Conf. on Machine Learning*. 102

James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., Levine, S., Hadsell, R., and Bousmalis, K. (2019). Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 30

Jayaraman, D., Gao, R., and Grauman, K. (2017). Unsupervised learning through one-shot image-based shape reconstruction. *ArXiv e-prints*, abs/1709.00505. 69

Jayaraman, D., Gao, R., and Grauman, K. (2018). Shapecodes: Self-supervised feature learning by lifting views to viewgrids. In *Proc. European Conf. on Computer Vision*. 23

Jefferson, G. (1949). The mind of mechanical man. *British Medical Journal*, 1(4616):1105–1110. 9

Ji, X., Henriques, J. F., and Vedaldi, A. (2018). Invariant information distillation for unsupervised image segmentation and clustering. *ArXiv e-prints*, abs/1807.06653. 12

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proc. Int'l Conf. on Machine Learning*. 12

Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux. 10

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018). End-to-end recovery of human shape and pose. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 25

Kennedy, J. and Eberhart, R. C. (1995). Particle Swarm Optimization. In *Proc. IEEE Int'l Conf. on Neural Networks*. 36, 49

Keskin, C., Kiraç, F., Kara, Y. E., and Akarun, L. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proc. European Conf. on Computer Vision*. 42, 67, 92

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 47

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proc. Int'l Conf. on Learning Representations*. 75, 102

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proc. Int'l Conf. on Learning Representations*. 94

Kirby, M. and Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103–108. 15, 79

Kocabas, M., Karagoz, S., and Akbas, E. (2019). Self-supervised learning of 3d human pose using multi-view geometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 22

Koller, D., Daniilidis, K., and Nagel, H. (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *Int'l Journal of Computer Vision*, 10(3):257–281. 32

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models - Principles and Techniques*. MIT Press. 42

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159. 32

Krähenbühl, P. (2018). Free supervision from video games. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 29, 31

Krähenbühl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *Proc. Neural Information Processing Systems*. 42, 43

Krähenbühl, P. and Koltun, V. (2013). Parameter learning and convergent inference for dense random fields. In *Proc. Int'l Conf. on Machine Learning*. 43

Krejov, P., Gilbert, A., and Bowden, R. (2017). Guided optimisation through classification and regression for hand pose estimation. *Computer Vision and Image Understanding*, 155:124–138. 38, 65, 75

Krogel, M. and Scheffer, T. (2004). Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57(1-2):61–81. 22

Kuch, J. J. and Huang, T. S. (1995). Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *Proc. IEEE Int'l Conf. on Computer Vision*. 3

Kuss, M. and Graepel, T. (2003). The geometry of kernel canonical correlation analysis. Technical Report 108, Max Planck Institute for Biological Cybernetics, Tübingen. 22

Kuznietsov, Y., Stückler, J., and Leibe, B. (2017). Semi-supervised deep learning for monocular depth map prediction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 23, 69

Kyriazis, N. and Argyros, A. A. (2013). Physically plausible 3d scene tracking: The single actor hypothesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 36, 49

Lai, P. L. and Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. In *Proc. IEEE-INNS-ENNS Int'l Joint Conf. on Neural Networks*. 22

Laine, S. and Aila, T. (2017). Temporal ensembling for semi-supervised learning. In *Proc. Int'l Conf. on Learning Representations*. 12, 13

Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *Proc. European Conf. on Computer Vision*. 68

Larsson, G., Maire, M., and Shakhnarovich, G. (2017). Colorization as a proxy task for visual understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 17, 20

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proc. Int'l Conf. on Machine Learning Workshops*. 12

Lee, J. and Kunii, T. L. (1993). Constraint-based hand animation. In Thalmann, N. M. and Thalmann, D., editors, *Models and Techniques in Computer Animation*. Springer Tokyo. 34

Leistner, C., Roth, P. M., Grabner, H., Bischof, H., Starzacher, A., and Rinner, B. (2008). Visual on-line learning in distributed camera networks. In *Proc. Int'l Conf. on Distributed Smart Cameras*. 22

Li, C., Zia, M. Z., Tran, Q., Yu, X., Hager, G. D., and Chandraker, M. (2019). Deep supervision with intermediate concepts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843. 28

Li, Y., Yang, M., and Zhang, Z. (2018). A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–20. 21, 22

Liang, J., Jiang, L., Niebles, J. C., Hauptmann, A. G., and Fei-Fei, L. (2019). Peeking into the future: Predicting future person activities and locations in videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 19

Lin, J. Y., Wu, Y., and Huang, T. S. (2000). Modeling the constraints of human hand motion. In *Proc. Workshop on Human Motion*. 34, 47, 72, 81

Liu, C., Yuen, J., and Torralba, A. (2011). SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):978–994. 21

Liu, J. and Mian, A. (2017). Learning human pose models from synthesized data for robust RGB-D action recognition. *ArXiv e-prints*, abs/1707.00823. 30, 93

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137. 12

Long, J., Shelhamer, E., and Darrell, T. (2015a). Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 28

Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015b). Learning transferable features with deep adaptation networks. In *Proc. Int'l Conf. on Machine Learning*. 28

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. Int'l Conf. on Machine Learning Workshops*. 75, 102

Madadi, M., Escalera, S., Baro, X., and Gonzalez, J. (2017). End-to-end global to local CNN learning for hand pose recovery in depth data. *ArXiv e-prints*, abs/1705.09606. 93

Makris, A. and Argyros, A. A. (2015). Model-based 3D hand tracking with on-line shape adaptation. In *Proc. British Machine Vision Conf.* 115

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks. In *Proc. IEEE Int'l Conf. on Computer Vision*. 74, 100, 101

Markman, E. M. (1991). *Categorization and naming in children. Problems of induction.* MIT Press, Cambridge, MA. 10

Massa, F., Russell, B. C., and Aubry, M. (2016). Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 10, 31, 91, 96, 97

Mathieu, M., Couprie, C., and LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. In *Proc. Int'l Conf. on Learning Representations*. 19, 21

Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., and Brox, T. (2018). What makes good synthetic training data for learning disparity and optical flow estimation? *Int'l Journal of Computer Vision*, 126(9):942–960. 29

McCarthy, J. and Feigenbaum, E. A. (1990). In memoriam: Arthur Samuel: Pioneer in machine learning. *AI Magazine*, 11(3):10–11. 9

McCorduck, P. (2004). *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence.* AK Peters Ltd. 9

Melax, S., Keselman, L., and Orsten, S. (2013). Dynamics based 3d skeletal hand tracking. In *Proc. Graphics Interface*. 42, 67

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proc. Int'l Conf. on Learning Representations Workshops*. 14

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proc. Neural Information Processing Systems*. 14, 15, 21

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *ArXiv e-prints*, abs/1411.1784. 75

Misra, I., Zitnick, C. L., and Hebert, M. (2016). Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. European Conf. on Computer Vision*. 19, 20

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill. 9

Moon, G., Yong Chang, J., and Mu Lee, K. (2018). V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 103, 105

Morerio, P., Cavazza, J., and Murino, V. (2018). Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *Proc. Int'l Conf. on Learning Representations*. 28

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 30, 93

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 *Proc. IEEE Int'l Conf. on Computer Vision*. 67

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proc. Int'l Conf. on Machine Learning*. 75

Nawrot, E., Mayo, S. L., and Nawrot, M. (2009). The development of depth perception from motion parallax in infancy. *Perception & Psychophysics*, 71(1):194–199. 10

Neumann, L., Zisserman, A., and Vedaldi, A. (2019). Future event prediction: If and when. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. 19

Neverova, N., Wolf, C., Nebout, F., and Taylor, G. W. (2015). Hand pose estimation through weakly-supervised learning of a rich intermediate representation. *ArXiv e-prints*, abs/1511.06728. 69

Neverova, N., Wolf, C., Nebout, F., and Taylor, G. W. (2017). Hand pose estimation through semi-supervised and weakly-supervised learning. *Computer Vision and Image Understanding*, 164:56–67. 91, 105

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. European Conf. on Computer Vision*. 10, 16, 68

Noroozi, M., Pirsiavash, H., and Favaro, P. (2017). Representation learning by learning to count. In *Proc. IEEE Int'l Conf. on Computer Vision*. 72, 77, 79

Oberweger, M. and Lepetit, V. (2017). DeepPrior++: Improving fast and accurate 3d hand pose estimation. In *Proc. IEEE Int'l Conf. on Computer Vision Workshops*. 67, 86, 87, 88, 101, 105

Oberweger, M., Riegler, G., Wohlhart, P., and Lepetit, V. (2016). Efficiently creating 3d training data for fine hand pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 68, 77

Oberweger, M., Wohlhart, P., and Lepetit, V. (2015a). Hands deep in deep learning for hand pose estimation. In *Proc. Computer Vision Winter Workshop*. 80, 85, 87

Oberweger, M., Wohlhart, P., and Lepetit, V. (2015b). Training a feedback loop for hand pose estimation. In *Proc. IEEE Int'l Conf. on Computer Vision*. 65, 69, 70

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2010). Markerless and efficient 26-DOF hand pose recovery. In *Proc. Asian Conf. on Computer Vision*. 35, 36, 42

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011a). Efficient model-based 3d tracking of hand articulations using Kinect. In *Proc. British Machine Vision Conf.* 2, 35, 37, 47, 49, 53, 54, 67

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 *Proc. IEEE Int'l Conf. on Computer Vision*. 37, 103

Oikonomidis, I., Lourakis, M. I. A., and Argyros, A. A. (2014). Evolutionary quasi-random search for hand articulations tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 36

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. J. (2018). Realistic evaluation of deep semi-supervised learning algorithms. In *Proc. Neural Information Processing Systems*. 13

Ostrovsky, Y., Meyers, E., Ganesh, S., Mathur, U., and Sinha, P. (2009). Visual parsing after recovery from blindness. *Psychological Science*, 20(12):1484–1491. 18

Owens, A., Wu, J., McDermott, J. H., Freeman, W. T., and Torralba, A. (2016). Ambient sound provides supervision for visual learning. In *Proc. European Conf. on Computer Vision*. 10, 21, 68

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359. 25, 26, 27, 28, 29

Panteleris, P., Oikonomidis, I., and Argyros, A. A. (2018). Using a single RGB frame for real time 3D hand pose estimation in the wild. In *Proc. IEEE Winter Conf. on Applications of Computer Vision*. 4, 93

Parameswaran, S. and Weinberger, K. Q. (2010). Large margin multi-task metric learning. In *Proc. Neural Information Processing Systems*. 28

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 14, 17, 20, 68, 79, 81

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017). Harvesting multiple views for marker-less 3d human pose annotations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 22

Pavllo, D., Feichtenhofer, C., Grangier, D., and Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 24

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572. 79

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics*. 15

Planche, B., Wu, Z., Ma, K., Sun, S., Kluckner, S., Lehmann, O., Chen, T., Hutter, A., Zakharov, S., Kosch, H., and Ernst, J. (2017). Depthsynth: Real-time realistic synthetic data generation from CAD models for 2.5d recognition. In *Proc. IEEE Int'l Conf. on 3D Vision*. 30

Poier, G., Opitz, M., Schinagl, D., and Bischof, H. (2019). MURAUER: Mapping unlabeled real data for label austerity. In *Proc. IEEE Winter Conf. on Applications of Computer Vision*. 5

Poier, G., Roditakis, K., Schulter, S., Michel, D., Bischof, H., and Argyros, A. A. (2015). Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. In *Proc. British Machine Vision Conf.* 5, 41, 62

Poier, G., Schinagl, D., and Bischof, H. (2018). Learning pose specific representations by predicting different views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 5, 23, 67, 79, 104

Poier, G., Schulter, S., Sternig, S., Roth, P. M., and Bischof, H. (2014). Hough forests revisited: An approach to multiple instance tracking from multiple cameras. In *Proc. German Conf. on Pattern Recognition*. 22

Poudel, R. P. K., Fonseca, J. A. S., Zhang, J. J., and Nait-Charif, H. (2013). A unified framework for 3d hand tracking. In *Proc. Int'l Symposium on Visual Computing*. 43, 45

Pratt, L. Y. (1992). Discriminability-based transfer between neural networks. In *Proc. Neural Information Processing Systems*. 10

Prinz, D. (1988). Robot chess. In Levy, D., editor, *Computer Chess Compendium*. Springer New York, NY. 9

Qian, C., Sun, X., Wei, Y., Tang, X., and Sun, J. (2014). Realtime and robust hand tracking from depth. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 35, 37, 38, 44, 45, 47, 49

Rad, M., Oberweger, M., and Lepetit, V. (2018a). Domain transfer for 3d pose estimation from color images without manual annotations. In *Proc. Asian Conf. on Computer Vision*. 31

Rad, M., Oberweger, M., and Lepetit, V. (2018b). Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 31, 91, 93, 94, 96, 97, 101, 102, 103, 105

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. Int'l Conf. on Learning Representations*. 74, 75, 102

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Technical report, OpenAI. 15, 19

Ranftl, R. and Pock, T. (2014). A deep variational model for image segmentation. In *Proc. German Conf. on Pattern Recognition*. 39

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. (2014). Video (language) modeling: a baseline for generative models of natural videos. *ArXiv e-prints*, abs/1412.6604. 19

Rehg, J. M. and Kanade, T. (1994). Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Proc. European Conf. on Computer Vision*. 35, 37

Rehg, J. M. and Kanade, T. (1995). Model-based tracking of self-occluding articulated objects. In *Proc. IEEE Int'l Conf. on Computer Vision*. 3

Rhodin, H., Salzmann, M., and Fua, P. (2018a). Unsupervised geometry-aware representation for 3d human pose estimation. In *Proc. European Conf. on Computer Vision*. 23

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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 22

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In *Proc. European Conf. on Computer Vision*. 29, 31

Roditakis, K. and Argyros, A. A. (2015). Quantifying the effect of a colored glove in the 3d tracking of a human hand. In *Proc. Int'l Conf. on Computer Vision Systems*. 38

Roditakis, K., Makris, A., and Argyros, A. A. (2017). Generative 3d hand tracking with spatially constrained pose sampling. In *Proc. British Machine Vision Conf.* 67

Rosales, R. and Sclaroff, S. (2006). Combining generative and discriminative models in a framework for articulated pose estimation. *Int'l Journal of Computer Vision*, 67(3):251–276. 29, 38

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int'l Journal of Computer Vision*, pages 1–42. 16

Savva, M., Chang, A. X., and Hanrahan, P. (2015). Semantically-enriched 3d models for common-sense knowledge. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. 29

Schmidhuber, J. and Prelinger, D. (1993). Discovering predictable classifications. *Neural Computation*, 5(4):625–635. 13, 14

Schmidt, T., Newcombe, R. A., and Fox, D. (2015). DART: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 39(3):239–258. 36

Schröder, M., Maycock, J., Ritter, H. J., and Botsch, M. (2014). Real-time hand tracking using synergistic inverse kinematics. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*. 35

Schulter, S., Leistner, C., Roth, P. M., Van Gool, L., and Bischof, H. (2011). On-line Hough forests. In *Proc. British Machine Vision Conf.* 46

Sharp, T., Keskin, C., Robertson, D. P., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A. W., and Izadi, S. (2015). Accurate, robust, and flexible real-time hand tracking. In *Proc. ACM Conf. on Human Factors in Computing Systems*. 4, 38, 47, 49, 93

Sheerman-Chase, T., Ong, E., and Bowden, R. (2013). Non-linear predictors for facial feature tracking across pose and expression. In *Proc. IEEE Int'l Conf. on Automatic Face and Gesture Recognition*. 70

Shotton, J., Girshick, R. B., Fitzgibbon, A. W., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. (2013). Efficient human pose estimation from single depth images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840. 43, 45, 46, 47, 51

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 30, 93

Simon, T., Joo, H., Matthews, I. A., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 22

Spelke, E. S. and Kinzler, K. D. (2007). Core knowledge. *Developmental Science*, 10:89–96. 32

Spurr, A., Song, J., Park, S., and Hilliges, O. (2018). Cross-modal deep variational hand pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 23, 94

Sridhar, S. (2016). *Tracking Hands in Action for Gesture-based Computer Input*. PhD thesis, Saarland University. 33, 35

Sridhar, S., Mueller, F., Oulasvirta, A., and Theobalt, C. (2015). Fast and robust hand tracking using detection-guided optimization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 38

Sridhar, S., Oulasvirta, A., and Theobalt, C. (2013). Interactive markerless articulated hand motion tracking using RGB and depth data. In *Proc. IEEE Int'l Conf. on Computer Vision*. 35

Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised learning of video representations using lstms. In *Proc. Int'l Conf. on Machine Learning*. 19

Stenger, B., Mendonça, P. R. S., and Cipolla, R. (2001). Model-based 3d tracking of an articulated hand. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 35

Strachey, C. S. (1952). Logical or non-mathematical programmes. In *Proc. ACM National Conference*. 9

Sudderth, E. B., Mandel, M. I., Freeman, W. T., and Willsky, A. S. (2004). Visual hand tracking using nonparametric belief propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. 34

Sümer, Ö., Dencker, T., and Ommer, B. (2017). Self-supervised learning of pose embeddings from spatiotemporal relations in videos. In *Proc. IEEE Int'l Conf. on Computer Vision*. 10

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017a). Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. IEEE Int'l Conf. on Computer Vision*. 116

Sun, S. (2013). A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038. 22

Sun, X., Shang, J., Liang, S., and Wei, Y. (2017b). Compositional human pose regression. In *Proc. IEEE Int'l Conf. on Computer Vision*. 24

Sun, X., Wei, Y., Liang, S., Tang, X., and Sun, J. (2015). Cascaded hand pose regression. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 65, 68, 93, 103

Sungatullina, D., Zakharov, E., Ulyanov, D., and Lempitsky, V. (2018). Image manipulation with perceptual discriminators. In *Proc. European Conf. on Computer Vision*. 30

Supancic, J. S., Rogez, G., Yang, Y., Shotton, J., and Ramanan, D. (2015). Depth-based hand pose estimation: Data, methods, and challenges. In *Proc. IEEE Int'l Conf. on Computer Vision*. 2, 65

Suwajanakorn, S., Snavely, N., Tompson, J., and Norouzi, M. (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Proc. Neural Information Processing Systems*. 22

Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., and Pauly, M. (2015). Robust articulated-ICP for real-time hand tracking. *Computer Graphics Forum*, 34(5):101–114. 36, 37

Tan, D. J., Cashman, T. J., Taylor, J., Fitzgibbon, A. W., Tarlow, D., Khamis, S., Izadi, S., and Shotton, J. (2016). Fits like a glove: Rapid and reliable hand shape personalization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 115

Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 42, 53, 54, 55, 67, 68, 75, 93

Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.-K., and Shotton, J. (2015). Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proc. IEEE Int'l Conf. on Computer Vision*. 65

Tang, D., Yu, T.-H., and Kim, T.-K. (2013). Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proc. IEEE Int'l Conf. on Computer Vision*. 43, 45, 69, 93

Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proc. Neural Information Processing Systems*. 13

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2016). Multi-view 3d models from single images with a convolutional network. In *Proc. European Conf. on Computer Vision*. 23, 29, 30, 69

Taylor, J., Bordeaux, L., Cashman, T. J., Corish, B., Keskin, C., Sharp, T., Soto, E., Sweeney, D., Valentin, J. P. C., Luff, B., Topalian, A., Wood, E., Khamis, S., Kohli, P., Izadi, S., Banks, R., Fitzgibbon, A. W., and Shotton, J. (2016). Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. on Graphics*, 35(4):143:1–143:12. 2, 35, 37, 38, 65, 67

Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A. W. (2012). The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 42, 77

Taylor, J., Stebbing, R. V., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., and Fitzgibbon, A. W. (2014). User-specific hand modeling from monocular depth sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 47, 115

Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., and Fua, P. (2016). Structured prediction of 3d human pose with deep neural networks. In *Proc. British Machine Vision Conf.* 81

Thrun, S. and Pratt, L. Y., editors (1998). *Learning to Learn*. Springer. 26

Tian, Y., Krishnan, D., and Isola, P. (2019). Contrastive multiview coding. *ArXiv e-prints*, abs/1906.05849. 21

Tkach, A., Tagliasacchi, A., Remelli, E., Pauly, M., and Fitzgibbon, A. W. (2017). Online generative model personalization for hand tracking. *ACM Trans. on Graphics*, 36(6):243:1–243:11. 115

Tompson, J., Stein, M., LeCun, Y., and Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. on Graphics*, 33(5):169:1–169:10. 29, 34, 36, 45, 49, 50, 51, 53, 55, 56, 57, 68, 76, 92, 93, 103

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. (2017). Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 69

Tung, H.-Y. F., Tung, H.-W., Yumer, E., and Fragkiadaki, K. (2017). Self-supervised learning of motion capture. In *Proc. Neural Information Processing Systems*. 25, 37

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460. 9, 10

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 28, 31

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *ArXiv e-prints*, abs/1412.3474. 28, 31

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. *Int'l Journal of Computer Vision*, 118(2):172–193. 38

Tzionas, D., Srikantha, A., Aponte, P., and Gall, J. (2014). Capturing hand motion with an RGB-D sensor, fusing a generative model with salient points. In *Proc. German Conf. on Pattern Recognition*. 44

van der Maaten, L. and Hinton, G. E. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605. 97, 108

van Tulder, G. and de Bruijne, M. (2019). Learning cross-modality representations from multi-modal images. *IEEE Transactions on Medical Imaging*, 38(2):638–648. 23

Vapnik, V. (1998). *Statistical learning theory*. Wiley. 12

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 29

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proc. Neural Information Processing Systems*. 15

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proc. Int'l Conf. on Machine Learning*. 16, 68

Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., and Murphy, K. (2018). Tracking emerges by colorizing videos. In *Proc. European Conf. on Computer Vision*. 20

Vyas, S., Rawat, Y. S., and Shah, M. (2018). Time-aware and view-aware video rendering for unsupervised representation learning. *ArXiv e-prints*, abs/1811.10699. 19, 23

Wan, C., Probst, T., Gool, L. V., and Yao, A. (2019). Self-supervised 3d hand pose estimation through training by fitting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 22, 24

Wan, C., Probst, T., Van Gool, L., and Yao, A. (2017). Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 69, 85, 86, 87, 91, 94, 96, 103, 104, 105

Wan, C., Probst, T., Van Gool, L., and Yao, A. (2018). Dense 3d regression for hand pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 105

Wang, G., Chen, X., Guo, H., and Zhang, C. (2018). Region ensemble network: Towards good practices for deep 3d hand pose estimation. *Journal of Visual Communication and Image Representation*, 55:404 – 414. 105

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 18

Wang, K., Lin, L., Jiang, C., Qian, C., and Wei, P. (2019a). 3d human pose machines with self-supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. (to be published). 24

Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153. 26, 28

Wang, W., Arora, R., Livescu, K., and Bilmes, J. A. (2015). On deep multi-view representation learning. In *Proc. Int'l Conf. on Machine Learning*. 22

Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proc. IEEE Int'l Conf. on Computer Vision*. 10, 18, 68

Wang, X., Jabri, A., and Efros, A. A. (2019b). Learning correspondence from the cycle-consistency of time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 10, 19, 20

Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to use t-SNE effectively. *Distill*. 109

Wei, D., Lim, J. J., Zisserman, A., and Freeman, W. T. (2018). Learning and using the arrow of time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 20

Wei, X. K., Zhang, P., and Chai, J. (2012). Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. on Graphics*, 31(6):188:1–188:12. 4, 38

Wetzler, A., Slossberg, R., and Kimmel, R. (2015). Rule of thumb: Deep derotation for improved fingertip detection. In *Proc. British Machine Vision Conf.* 68

Wu, B., Zhou, X., Zhao, S., Yue, X., and Keutzer, K. (2018). SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *ArXiv e-prints*, abs/1809.08495. 31

Wu, Y. and Huang, T. S. (1999). Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. IEEE Int'l Conf. on Computer Vision*. 3, 49

Wu, Y., Lin, J. Y., and Huang, T. S. (2001). Capturing natural hand articulation. In *Proc. IEEE Int'l Conf. on Computer Vision*. 2, 42, 43, 67

Xie, J., Girshick, R., and Farhadi, A. (2016). Deep3D: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *Proc. European Conf. on Computer Vision*. 23, 69

Xiong, X. and De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 70

Xu, C. and Cheng, L. (2013). Efficient hand pose estimation from a single depth image. In *Proc. IEEE Int'l Conf. on Computer Vision*. 2, 34

Xu, C., Govindarajan, L. N., Zhang, Y., Stewart, J., Bichler, Z., Jesuthasan, S., Claridge-Chang, A., Mathuru, A. S., Tang, W., Zhu, P., and Cheng, L. (2017). Lie-X: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *Int'l Journal of Computer Vision*, 123(3):454–478. 105

Xu, C., Nanjappa, A., Zhang, X., and Cheng, L. (2015). Estimate hand poses efficiently from single depth images. *Int'l Journal of Computer Vision*. (available under Open Access). 42

Yamauchi, K., Oota, M., and Ishii, N. (1999). A self-supervised learning system for pattern recognition by sensory integration. *Neural Networks*, 12(10):1347–1358. 14

Yang, J., Reed, S. E., Yang, M., and Lee, H. (2015). Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Proc. Neural Information Processing Systems*. 69

Ye, M., Wang, X., Yang, R., Ren, L., and Pollefeys, M. (2011). Accurate 3d pose estimation from a single depth image. In *Proc. IEEE Int'l Conf. on Computer Vision*. 42

Ye, Q., Yuan, S., and Kim, T.-K. (2016). Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In *Proc. European Conf. on Computer Vision*. 4, 93

Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Yong Chang, J., Mu Lee, K., 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. A., and Kim, T.-K. (2018). Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 3

Yuan, S., Ye, Q., Stenger, B., Jain, S., and Kim, T.-K. (2017). Big Hand 2.2M benchmark: Hand pose data set and state of the art analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 68, 86, 93

Zamir, A. R., Wekel, T., Agrawal, P., Wei, C., Malik, J., and Savarese, S. (2016). Generic 3d representation via pose estimation and matching. In *Proc. European Conf. on Computer Vision*. 68

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *Proc. European Conf. on Computer Vision*. 17, 20, 77

Zhang, R., Isola, P., and Efros, A. A. (2017). Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 17, 68, 79

Zhao, M., Li, T., Abu Alsheikh, M., Tian, Y., Zhao, H., Torralba, A., and Katabi, D. (2018). Through-wall human pose estimation using radio signals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 27

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017a). Unsupervised learning of depth and ego-motion from video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 23, 69

Zhou, X., Huang, Q., Sun, X., Xue, X., and Wei, Y. (2017b). Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proc. IEEE Int'l Conf. on Computer Vision*. 85

Zhou, X., Wan, Q., Zhang, W., Xue, X., and Wei, Y. (2016). Model-based deep hand pose estimation. In *Proc. Int'l Joint Conf. on Artificial Intelligence*. 24, 38, 67

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE Int'l Conf. on Computer Vision*. 30

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. Int'l Conf. on Machine Learning*. 12

Zimmermann, C. and Brox, T. (2017). Learning to estimate 3d hand pose from single RGB images. In *Proc. IEEE Int'l Conf. on Computer Vision*. 29