Hand-Object Contact Force Estimation from Markerless Visual Tracking

Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A. Argyros, and Abderrahmane Kheddar, *Senior Member, IEEE*

Abstract—We consider the problem of estimating realistic contact forces during manipulation, backed with ground-truth measurements, using vision alone. Interaction forces are usually measured by mounting force transducers onto the manipulated objects or the hands. Those are costly, cumbersome, and alter the objects' physical properties and their perception by the human sense of touch. Our work establishes that interaction forces can be estimated in a cost-effective, reliable, non-intrusive way using vision. This is a complex and challenging problem. Indeed, in multi-contact, a given motion can generally be caused by an infinity of possible force distributions. To alleviate the limitations of traditional models based on inverse optimization, we collect and release the first large-scale dataset on manipulation kinodynamics as 3.2 hours of synchronized force and motion measurements under 193 object-grasp configurations. We learn a mapping between high-level kinematic features based on the equations of motion and the underlying manipulation forces using recurrent neural networks (RNN). The RNN predictions are consistently refined using physics-based optimization through second-order cone programming (SOCP). We show that our method can successfully capture interaction forces compatible with both the observations and the way humans intuitively manipulate objects, using a single RGB-D camera.

Index Terms—Force sensing from vision, hand-object tracking, manipulation, pattern analysis, sensors, tracking

1 Introduction

Haptics is of fundamental importance to interact with objects and perceive their physical and functional properties. Recent work has showed how the latter could be inferred from vision [1], [2], [3]. In contrast, human manipulation remains little understood at the level of the underlying interaction forces, which are traditionally measured using force transducers. The latter are costly, cumbersome, and intrusive on both the object and the human haptic sense. Recent advances in markerless visual tracking enabled the non-intrusive monitoring of hand-object motions. Computer vision techniques would thus be an ideal substitute for current force sensing technologies.

This is an extremely challenging perspective, as tracking hand-object interactions is difficult due to strong mutual occlusions. Even when object and motion are fully known, the force estimation problem is ill-posed or indeterminate in multi-contact, i.e., there exists an infinity of compatible force distributions (e.g., of varying grip strengths). While it is possible to compute physically plausible forces, capturing the real forces applied is an open problem in multiple fields (Section 2). Kinesiology research produced successful attempts at modeling grip forces by inverse optimization, e.g., during static prehension [4] or two-finger circular motion [5]. Albeit of limited scope, these suggest that it may be possible to construct a general model on human grasping, provided a rich dataset on manipulation kinodynamics (motion and forces). In our work, we show that physics-based optimization can be used together with learning to capture manipulation forces with a single RGB-D camera.

- We construct the first large-scale dataset on human manipulation kinodynamics, containing 3.2 hours of high-frequency measurements under 193 different object-grasp configurations (Section 3).
- We propose a force estimation framework that relies simultaneously on a recurrent neural network to predict forces that are consistent with the way humans naturally manipulate objects, and on a second-order cone program (SOCP) guaranteeing the physical correctness of the final force distribution (Section 4).
- We thoroughly validate our approach on ground-truth measurements (Section 5) and show that it can seamlessly be extended to visual tracking (Section 6).

Our dataset is dedicated to static prismatic grasps, i.e., with the thumb in direct opposition to antagonist fingers. We discuss instrumentation limitations and show that the optimization-learning framework can still address scenarios beyond the focus of our study (Section 7). Finally, we thoroughly discuss the current limitations, extensions and applications of our work (Section 8). A preliminary version of this research, estimating normal forces from vision, appeared in [6]. Our current study extends the latter idea and includes an improved formulation of the optimization and learning models accounting for 3D time-coherent forces, as well as

- *T.-H. Pham and A. Kheddar are with the CNRS-AIST Joint Robotics Laboratory, UMI3218/RL Tsukuba Central 1, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560 JAPAN, and with CNRS-University of Montpellier, LIRMM IDH, UMR5506, Montpellier 34090, France. E-mail: {pham.main, kheddar}@gmail.com.*
- *N. Kyriazis is with the Institute of Computer Science, FORTH, Heraklion GR-700 13, Greece. E-mail: kyriazis@ics.forth.gr.*
- *A. A. Argyros is with the Institute of Computer Science, FORTH, Heraklion GR-700 13, Greece and with the Computer Science Department, University of Crete, Heraklion, Rethimno 741 00, Greece. E-mail: argyros@ics.forth.gr.*

Manuscript received 24 Aug. 2016; revised 29 Aug. 2017; accepted 18 Sept. 2017. Date of publication 26 Oct. 2017; date of current version 1 Nov. 2018. (Corresponding author: Tu-Hoa Pham.)
Recommended for acceptance by R. Bowden.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2759736

algorithmic descriptions and extensive validation experiments that have not been presented before. To foster the research in this new topic, we make the manipulation kinodynamics dataset publicly available.[1]

2 Related Work

Monitoring Hand-Object Interactions. Current force transduction and sensing technologies are costly and require frequent calibration. Mounting them onto objects biases physical properties such as shape, mass distribution and friction, while mounting them onto hands obstructs the human haptic sense, limiting the natural range of motion. In contrast, there is evidence that fingertip forces can be correlated to changes in the coloration of fingernails and surrounding skin [7], [8], [9]. The latter setups already suggest that computer vision can measure touch forces.

In conjunction with force transducers, marker-based motion capture was used in [10] to estimate hand joint compliance and synthesize interaction animations. Motion capture markers being arguably invasive and difficult to deploy in daily activities, the topic of markerless hand tracking was introduced in [11] and lately received renewed attention in [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. During manipulation, hand-object interactions cause mutual occlusions that generative approaches can employ to enforce priors in the optimization process [31], [32], [33], [34]. In particular, force models were used to select hand poses that are compatible with the observations through physical simulation [35], [36], [37]. In contrast with our approach, these models only need to capture physically plausible distributions rather than the actual forces being applied, which may substantially differ.

Biomechanical Models for Human Prehension. Prehension is an active research topic in the kinesiology field; an interest that stems from the remarkable dexterity and complexity of the human hand. Thus, inverse optimization approaches for manipulation have mostly resulted in models that, albeit sophisticated, rely on rather strong simplifications such as on the motion's dimensionality, e.g., static prehension [4]. Other approaches allow limited motion, e.g., circular [5], using a simplified grasp model in which individual fingers and hand surfaces are grouped into functional units named virtual fingers [38]. A hand holding a cup is thus seen as the thumb on one side and a virtual finger on the opposite side realizing the total wrench due to the four antagonist fingers. Under this formalism, the five-finger grasp is effectively seen as *two-finger*, and the knowledge of one force fully determines the other. In reality, the force distribution problem is generally indeterminate as five-finger forces can compensate each other to cause the same motion.

The virtual finger model was also applied on nominal-internal forces during 1D translational motions [39]. Internal forces represent the set of forces that humans apply in excess to the nominal forces that are required to create a given motion [40], [41]. For instance, when holding a cup statically, nominal forces directly compensate gravity, while internal forces secure the object through a firm grip but cancel each other out [42], [43]. Past studies showed that humans control internal forces to prevent slip, muscle fatigue or damaging fragile objects [44], [45], [46]. Overall, in reviewing several optimization-based models attempting to predict muscle activation patterns, [47] showed that the high redundancy of the human body makes it particularly difficult to identify clear optimization criteria in the way the central nervous system regulates human efforts at the musculoskeletal level.

Force Sensing From Vision. The force sensing from vision (FSV) framework we present is a continuation of our earlier work in [6], that was limited to 1D normal force measurements, four-finger grasps and relatively limited experimental conditions. Normal forces on four-finger grasps were also estimated in the recent work of [48]. In contrast, our present work is grounded in a new dataset of 3D force measurements on five-finger, diverse manipulation experiments. Our past work used shallow multilayer perceptrons (MLP) to learn internal forces. Such an approach is difficult to generalize as the decomposition into nominal and internal components is not intrinsic, but rather depends on the objective function chosen to minimize nominal forces. While the extended approach we present here still builds upon the formulation of the force distribution problem as a second-order cone program (SOCP) [49], [50], we also capitalize on the recent success of deep learning applications to manipulation and monitoring of human activities [2], [51], [52] to construct a network that directly learns full 3D manipulation forces, avoiding the need for arbitrary constraints and hand-engineering [53]. Our work is also inspired by [54], which estimated ground reaction forces from motion capture using a damped spring model. Recently, contact forces were computed for deformable objects [55] and conversely by considering the human body elastic [56]. [19] showed that forces play a crucial role to understand hand-object interactions from vision and noted the challenge of obtaining ground-truth contact points and forces humans use instinctively, which we address in our work.

3 Manipulation Kinodynamics Dataset

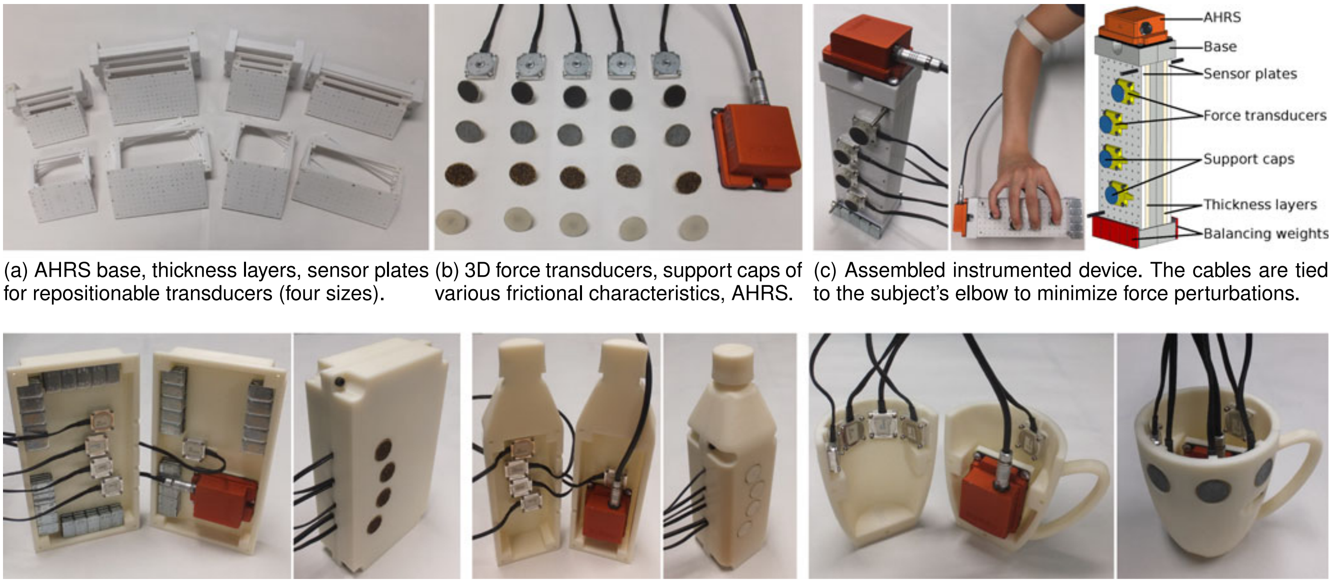
Public datasets have benefitted multiple research topics such as scene understanding [57], [58], visual tracking [17], [59] and robotic grasping [60], [61]. In contrast, datasets viewing manipulation not only from the angle of vision but also of touch have been more scarce so far [10]. We introduce a new, extensive dataset on human manipulation kinodynamics.

3.1 Experimental Setup

While using real objects may initially seem ideal, instrumenting them with force and motion sensors is impractical, making data collection difficult and lengthy. Additionally, physical properties of arbitrary objects (e.g., inertia matrices) are seldom publicly available and must therefore be manually identified [62], [63]. Finally, the instrumentation may result in measured forces that substantially differ from those that would have been applied on the original objects.

Instead, we construct dedicated instrumented devices, pictured in Fig. 1. Two symmetric parts (for thumb and antagonist fingers) form a base holding an attitude and heading reference system (AHRS, Xsens MTi-300). Sensor plates are mounted on both sides, on which 3D precision force transducers (Tec Gihan USL06-H5-50N) can be positioned

[1] 1. https://github.com/jrl-umi3218/ManipulationKinodynamics.



(a) AHRS base, thickness layers, sensor plates for repositionable transducers (four sizes).

(b) 3D force transducers, support caps of various frictional characteristics, AHRS.

(c) Assembled instrumented device. The cables are tied to the subject's elbow to minimize force perturbations.

(d) 3D-printed box-shaped device with extra mass.

(e) Bottle-shaped device.

(f) Mug-shaped device for non-prismatic, spherical grasp.

Fig. 1. Instrumented devices of adjustable physical and grasping properties (a-c), or based on everyday objects to allow intuitive interactions (d-f).

by 8 mm steps on their surface. Thickness layers can be inserted to increase the grasp width by 5 mm increments, bringing its range between 46 mm and 86 mm. The force transducers are fitted with support caps of different surface textures: PET, sand paper of grit 40 (coarse), 150 (medium) and 320 (fine). The mass distribution can be adjusted with balancing weights inside and on the surface of the instrumented device. We 3D-print four sets of instrumented modules, with sensor plates of dimensions 80×152, 56×152, 80×96 and $56 \times 96\ \mathrm{mm}^2$. This setup allows collecting force and kinematics measurements under diverse grasp poses, friction conditions and mass distributions, obtained from the CAD models of the individual components.

3.2 The Dataset

Eleven right-handed volunteers (three females, eight males), took part in our experiments. Each subject performed series of eight manipulation experiments as follows. For each series, the subject was given an instrumented device of random shape, thickness and surface texture, with the AHRS either at the top or the bottom, and at random an additional 400 g mass inside. Before each trial, force transducers were placed at the subject's grasp preference. Each trial consisted in the subject grasping the object and manipulating it for approximately 60 s. Every 10 s, to ensure the diversity of motion and forces in the final dataset, the subject was given randomly chosen instructions on speed, direction and task (e.g., slow forward pouring motion, fast left and right oscillations). After each trial, a 50 g balancing weight was attached to a randomly chosen side, excluding sensor plates. Trials could be interrupted when the object became uncomfortable to manipulate. Throughout eight trials, we thus measured the effect of mass variations between 0 g and 350 g, or 400 g and 750 g with the additional internal mass, arranged differently across series.

Overall, we collect motion and force measurements for 3.2 hours of manipulation experiments under 193 conditions of motion, friction, mass distribution and grasp. Precisely, we counted 71 unique grasp shapes in terms of fingertip poses relative to each other and 193 unique grasps relative to the object's center of mass. For each experiment, we provide: the global orientation $\mathbf{q}$, rotational velocity $\boldsymbol{\omega}$ and translational acceleration $\mathbf{a}$ measured by the AHRS at 400 Hz; 3D force measurements expressed in the reference frame of the object $\mathcal{R}_{\text{obj.}}$, subsampled from 500 Hz to 400 Hz to match the AHRS; the physical properties of the object: mass m, inertia matrix $\mathbf{J}$ about the center of mass $\mathbf{C}$; and the grasp parameters: for each finger $k \in \mathcal{F}$, the friction coefficient μ_k at contact point $\mathbf{P}_k^c$, and $\mathcal{R}_k = \left(\mathbf{n}_k, \mathbf{t}_k^x, \mathbf{t}_k^y\right)$ a local right-handed reference frame with $\mathbf{n}_k$ the normal to the surface oriented from the finger to the object and $\left(\mathbf{t}_k^x, \mathbf{t}_k^y\right)$ two orthogonal vectors forming a basis of the contact tangential plane. Friction coefficients are estimated by instructing the subjects to press and pull the force transducers until slipping and computing the maximum ratio between tangential and normal forces through the Coulomb friction model

$$\left\| g_k \mathbf{t}_k^x + h_k \mathbf{t}_k^y \right\|_2 \leq \ \mu_k f_k, \tag{1}$$

with (f_k, g_k, h_k) the local decomposition of contact force $\mathbf{F}_k$

$$\mathbf{F}_k = f_k \mathbf{n}_k + g_k \mathbf{t}_k^x + h_k \mathbf{t}_k^y. \tag{2}$$

μ_k can also be measured geometrically as the tangent of the maximum angle an object can be tilted before sliding, enabling its characterization even without force transducers. Models dedicated to finger contacts [64] can also be considered for dextrous tasks, e.g., in-hand manipulation.

3.3 Equations of Motion and Synchronization

Let $\mathcal{F}_{\mathrm{c}}$ and $\boldsymbol{\tau}_c$ be the net force and torque due to individual contact forces, respectively, and $\mathcal{F}_{\mathrm{d}}$ and $\boldsymbol{\tau}_d$ the net force and torque due to non-contact forces. The Newton-Euler equations of rigid body motion at the center of mass are

$$\begin{cases} \mathcal{F}_{\mathrm{c}} & = m\mathbf{a} - \mathcal{F}_{\mathrm{d}} \\ \boldsymbol{\tau}_{\mathrm{c}} & = \mathbf{J}_{\mathbf{q}} \cdot \boldsymbol{\alpha} + \boldsymbol{\omega} \times (\mathbf{J}_{\mathbf{q}} \cdot \boldsymbol{\omega}) - \boldsymbol{\tau}_{\mathrm{d}}, \end{cases} \tag{3}$$

with $\mathbf{J_q}$ the inertia matrix at orientation $\mathbf{q}$ and $\boldsymbol{\alpha}$ the rotational acceleration of the object, obtained by numerical differentiation of the AHRS rotational velocity measurements $\boldsymbol{\omega}$. Typically, $\boldsymbol{\mathcal{F}}_{\text{d}} = m\mathbf{g}$ and $\boldsymbol{\tau}_{\text{d}} = \mathbf{0}$, with $\mathbf{g}$ the gravity vector ($\mathbf{g} \approx -9.81\mathbf{z}$, $\mathbf{z}$ the vertical vector pointing up in the world frame). Intuitively, Eq. (3) means that $\boldsymbol{\mathcal{F}}_{\text{c}}$ and $\boldsymbol{\tau}_c$ can be computed and as a function of purely kinematic terms, as well as from individual finger forces (see Eq. (10)).

By computing in $\mathcal{R}_{\text{obj.}}$ net contact forces $\boldsymbol{\mathcal{F}}_{\text{c}}^{\text{kin}}$ from AHRS kinematics and $\boldsymbol{\mathcal{F}}_{\text{c}}^{\text{dyn}}$ from force transducers, both are synchronized temporally by computing, by cross-correlation, the best delay ΔT such that $\boldsymbol{\mathcal{F}}_c^{\text{kin}}(t) = \boldsymbol{\mathcal{F}}_c^{\text{dyn}}(t + \Delta T)$. Second, both AHRS and force transducers are subject to measurement errors (manufacturer specifications: ± 0.3 m s^{-2} maximum AHRS translational acceleration error, ± 1 N maximum force error per transducer). Acceleration errors $\Delta\mathbf{a}$ result in net force errors $m\Delta\mathbf{a}$, e.g., ± 0.15 N for a 0.5 kg object. In contrast, individual force transducer errors can potentially add up to ± 5 N over five fingers. In practice, we measured an average net force discrepancy of 0.33 N between AHRS and force transducers across the dataset. For each experiment, we compute the average net force $\Delta\boldsymbol{\mathcal{F}}_{\text{c}}$ and torque $\Delta\boldsymbol{\tau}_{\text{c}}$ discrepancies between AHRS and force transducers signals. We align the latter (noisier) onto the former by computing offsets $(\Delta\mathbf{F}_k)_{k\in\mathcal{F}}$ that are minimal and best result in $\Delta\boldsymbol{\mathcal{F}}_{\text{c}}$ and $\Delta\boldsymbol{\tau}_{\text{c}}$, through three costs

$$\begin{cases} \mathcal{C}_{\text{var}}\big((\Delta\mathbf{F}_k)_k\big) = \sum_{k\in\mathcal{F}} \|\Delta\mathbf{F}_k\|_2^2 \\ \mathcal{C}_{\boldsymbol{\mathcal{F}}_{\text{c}}}\big((\Delta\mathbf{F}_k)_k\big) = \big\|\Delta\boldsymbol{\mathcal{F}}_{\text{c}} - \sum_{k\in\mathcal{F}} [\Delta\mathbf{F}_k]\big\|_2^2 \\ \mathcal{C}_{\boldsymbol{\tau}_{\text{c}}}\big((\Delta\mathbf{F}_k)_k\big) = \Big\|\Delta\boldsymbol{\tau}_{\text{c}} - \sum_{k\in\mathcal{F}} \Big[\overrightarrow{\mathbf{CP}_k} \times \Delta\mathbf{F}_k\Big]\Big\|_2^2 \end{cases} \tag{4}$$

We compute minimal force transducer offsets by solving

$$\min_{\Delta\mathbf{F}_k} \{\mathcal{C}_{\text{var}} + \mathcal{C}_{\boldsymbol{\mathcal{F}}_{\text{c}}} + \mathcal{C}_{\boldsymbol{\tau}_{\text{c}}}\} \tag{5}$$

In practice, we normalize $\mathcal{C}_{\boldsymbol{\mathcal{F}}_{\text{c}}}$ and $\mathcal{C}_{\boldsymbol{\tau}_{\text{c}}}$ with the initial discrepancies $\Delta\boldsymbol{\mathcal{F}}_{\text{c}}$ and $\Delta\boldsymbol{\tau}_{\text{c}}$, respectively. We solve Eq. (5) by sequential least squares programming [65] and correct the force transducer measurements with the resulting offsets.

4 Force Model

From Eq. (3), only net forces and torques are determined by the object's motion. We reconstruct individual finger forces by combining physics-based optimization and learning.

4.1 Physics-Based Optimization for Manipulation

We construct forces causing a given motion by solving a second-order cone program (SOCP) [49], [50] of the form

$$\min \quad \mathcal{C}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{r}^T\mathbf{x} \tag{6}$$

$$\text{s.t.} \qquad \big\|\mathbf{A}_j\mathbf{x} + \mathbf{b}_j\big\|_2 \leq \mathbf{c}_j^T\mathbf{x} + \mathbf{d}_j, \quad j = 1, \ldots, m \tag{7}$$

$$\mathbf{E}\mathbf{x} \ \leq \mathbf{f} \tag{8}$$

$$\mathbf{G}\mathbf{x} = \mathbf{h}, \tag{9}$$

with $\mathbf{x} = (f_1, g_1, h_1, \ldots, f_5, g_5, h_5)^T$ the force components of Eq. (2), taken as 15 optimization parameters. We now define constraint matrices $\mathbf{P}, \mathbf{r}, \mathbf{A}_j, \mathbf{b}_j, \mathbf{c}_j, \mathbf{d}_j, \mathbf{E}, \mathbf{f}, \mathbf{G}, \mathbf{h}$.

Positivity. With the convention that each contact normal $\mathbf{n}_k$ is oriented inwards the object, normal forces f_k are nonnegative: $\forall k \in \mathcal{F}, f_k \geq 0$. In Eq. (8), $\mathbf{E}$ is a selection matrix (with only 0 and 1) such that $\mathbf{E}\mathbf{x} = \big((f_k)_{k\in\mathcal{F}}\big)^T$ and $\mathbf{f} = \mathbf{0}$.

Friction. We define one friction constraint (see Eq. (1)) per finger, i.e., $m = 5$ in Eq. (7). $\forall k \in \mathcal{F}$, $\mathbf{A}_k$ is a selection matrix such that $\mathbf{A}_k\mathbf{x} = (g_k, h_k)^T$, $\mathbf{c}_k$ is a selection vector (with only 0 and μ_k) such that $\mathbf{c}_k^T\mathbf{x} = (\mu_k f_k)$, and $\mathbf{b}_k = \mathbf{d}_k = \mathbf{0}$.

Equations of Motion. From Eq. (3), we compute net contact force $\boldsymbol{\mathcal{F}}_{\text{c}}$ and torque $\boldsymbol{\tau}_c$ from kinematics only. By construction, $\boldsymbol{\mathcal{F}}_{\text{c}}$ and $\boldsymbol{\tau}_c$ are linked to individual finger forces by

$$\boldsymbol{\mathcal{F}}_{\text{c}} = \sum_{k\in\mathcal{F}} \mathbf{F}_k \quad \text{and} \quad \boldsymbol{\tau}_{\text{c}} = \sum_{k\in\mathcal{F}} \Big[\overrightarrow{\mathbf{CP}_k} \times \mathbf{F}_k\Big]. \tag{10}$$

We formulate 6 equality constraints by projecting Eq. (10) in the world frame $\mathcal{R}_{\text{W}} = (\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3})$. In Eq. (9), $\mathbf{G}$ and $\mathbf{h}$ are of respective sizes 6×15 and 6×1, with

$$\begin{gathered} \forall i = 1, \ldots, 3; \quad \forall j = 1, \ldots, 15; \quad \forall k = 1, \ldots, 5; \\ \mathbf{G}(i, j) = \begin{cases} \mathbf{n}_k \cdot \mathbf{v}_i & \text{if} \quad j = 3(k-1)+1 \\ \mathbf{t}_k^x \cdot \mathbf{v}_i & \text{if} \quad j = 3(k-1)+2 \\ \mathbf{t}_k^y \cdot \mathbf{v}_i & \text{if} \quad j = 3(k-1)+3 \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{G}(i+3, j) = \begin{cases} \Big[\overrightarrow{\mathbf{CP}_k} \times \mathbf{n}_k\Big] \cdot \mathbf{v}_i & \text{if } j = 3(k-1)+1 \\ \Big[\overrightarrow{\mathbf{CP}_k} \times \mathbf{t}_k^x\Big] \cdot \mathbf{v}_i & \text{if } j = 3(k-1)+2 \\ \Big[\overrightarrow{\mathbf{CP}_k} \times \mathbf{t}_k^y\Big] \cdot \mathbf{v}_i & \text{if } j = 3(k-1)+3 \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{h}(i, 1) = \boldsymbol{\mathcal{F}}_{\text{c}} \cdot \mathbf{v}_i \quad \text{and} \quad \mathbf{h}(i+3, 1) = \boldsymbol{\tau}_c \cdot \mathbf{v}_i. \end{gathered} \tag{11}$$

Cost. Physically plausible forces can be computed with a cost depending only on $\mathbf{x}$, e.g., minimal L^2 norm [6]

$$\mathcal{C}_{L^2}(\mathbf{x}) = \sum_{k\in\mathcal{F}} \big[{f_k}^2 + {g_k}^2 + {h_k}^2\big] = \sum_{k\in\mathcal{F}} \|\mathbf{F}_k\|_2^2. \tag{12}$$

Yet, the resulting forces can significantly differ from those humans actually apply (see Fig. 2). Instead, we consider a cost minimizing the discrepancy with given target forces $\widetilde{\mathbf{F}}_k$

$$\mathcal{C}_{\widetilde{\mathbf{F}}_k}(\mathbf{x}) = \sum_{k\in\mathcal{F}} \Big\|\mathbf{F}_k - \widetilde{\mathbf{F}}_k\Big\|_2^2. \tag{13}$$

In Section 4.2, we take $\widetilde{\mathbf{F}}_k$ as force transducer measurements to correct sensing uncertainties. From Section 5.1 onwards, target forces $\widetilde{\mathbf{F}}_k$ are neural network force predictions. We depict the SOCP force correction architecture in Fig. 3.

4.2 Learning Features

The dataset parameters of Section 3 fall into three categories:

- Object and grasp parameters: location of the center of mass $\mathbf{C}$ in $\mathcal{R}_{\text{obj.}}$, mass m, inertia matrix $\mathbf{J}$, contact point locations $\mathbf{P}_k$ and friction coefficients μ_k.
- Kinematic parameters: appearing in Eq. (3) are the object's orientation $\mathbf{q}$ in $\mathcal{R}_{\text{W}}$, rotational velocity $\boldsymbol{\omega}$, rotational acceleration $\boldsymbol{\alpha}$ and translational acceleration $\mathbf{a}$.

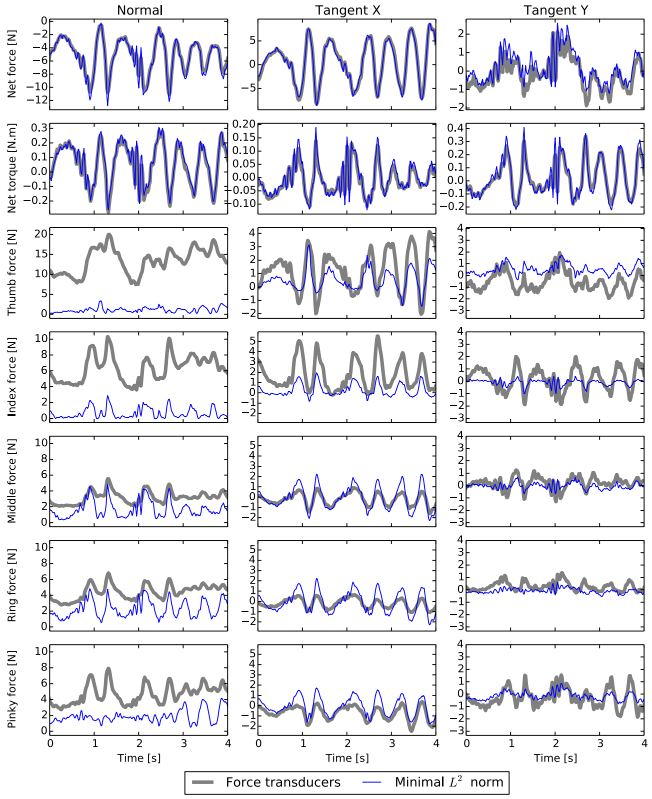


Fig. 2. Force distributions computed only by physics-based optimization are guaranteed to result in the observed motion (net force and torque) but can significantly differ from the real distributions at the finger level.

- $\mathbf{q}, \boldsymbol{\omega}, \mathbf{a}$ are directly measured by the AHRS. $\boldsymbol{\alpha}$ is obtained by simple numerical differentiation of $\boldsymbol{\omega}$. Alternatively, the relevant kinematic parameters can be obtained from visual tracking, through double differentiation of the object's pose and orientation.
- Force transducer measurements $\widetilde{\mathbf{F}}_k$.

To alleviate sensing uncertainties, we extract physically plausible force distributions $\mathbf{F}_k$ in the vicinity of the possibly inaccurate measurements $\widetilde{\mathbf{F}}_k$, as depicted in Fig. 3. The objective is then to learn $\mathbf{F}_k$ based on input parameters that depend only on the grasp, the object and its kinematics. We select these input features based on their contribution to the equations of motion. A first approach could be to take the raw parameters listed above. However, their influence is often determined not individually but rather in interaction with other parameters. From Eq. (10), the positions of the center of mass $\mathbf{C}$ and contact points $\mathbf{P}_k$ are meaningful not on their own but in relation to each other as $\overrightarrow{\mathbf{CP}}_k$. Similarly, from Eq. (3), we summarize the contributions of m, $\mathbf{a}$, $\mathbf{J}$, $\mathbf{q}$, $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$ into the target net contact force $\mathcal{F}_c$ and torque $\boldsymbol{\tau}_c$.

Recall that $\mathcal{F}_c$ and $\boldsymbol{\tau}_c$ are expressed in $\mathcal{R}_{\mathrm{W}}$. Since the dataset focuses on static grasps, for each experiment, the contact points are constant in any frame attached to the object. We account for translational and rotational invariances by projecting $\mathcal{F}_c$, $\boldsymbol{\tau}_c$ and $\overrightarrow{\mathbf{CP}}_k$ on $\mathcal{R}_{\mathrm{obj.}}$. Thus, the input features stemming from the Newton-Euler equations are

$$\forall (k, \mathbf{v}) \in \mathcal{F} \times \mathcal{R}_{\mathrm{obj.}}, \quad \begin{cases} p_{\mathbf{v}}^{\mathcal{F}_c} & = & \mathcal{F}_c \cdot \mathbf{v} \\ p_{\mathbf{v}}^{\boldsymbol{\tau}_c} & = & \boldsymbol{\tau}_c \cdot \mathbf{v} \\ p_{\mathbf{v}}^{\mathbf{P}_k} & = & \overrightarrow{\mathbf{CP}}_k \cdot \mathbf{v}. \end{cases} \tag{14}$$

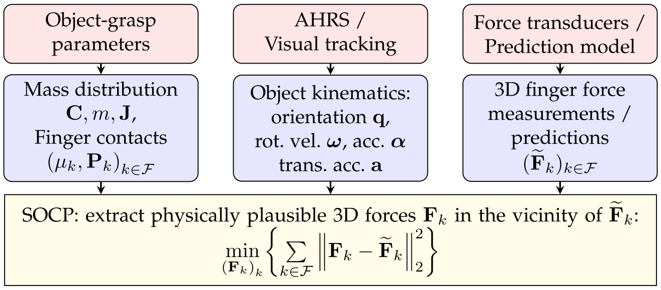

Object-grasp parameters
AHRS / Visual tracking
Force transducers / Prediction model
Mass distribution $\mathbf{C}, m, \mathbf{J}$,
Finger contacts $(\mu_k, \mathbf{P}_k)_{k\in\mathcal{F}}$
Object kinematics: orientation $\mathbf{q}$, rot. vel. $\boldsymbol{\omega}$, acc. $\boldsymbol{\alpha}$ trans. acc. $\mathbf{a}$
3D finger force measurements / predictions $(\widetilde{\mathbf{F}}_k)_{k\in\mathcal{F}}$
SOCP: extract physically plausible 3D forces $\mathbf{F}_k$ in the vicinity of $\widetilde{\mathbf{F}}_k$:
$\min_{(\mathbf{F}_k)_k} \left\{ \sum_{k\in\mathcal{F}} \left\| \mathbf{F}_k - \widetilde{\mathbf{F}}_k \right\|_2^2 \right\}$

Fig. 3. By SOCP, we extract force distributions compatible with the observed motion in the vicinity of target forces (measured or predicted).

With $p^{\mu} = <\mu_k>_{k\in\mathcal{F}}$ the average friction coefficient, we regroup these parameters, derived from the grasp-object properties and kinematics, into a 22-element vector $\mathbf{K}$

$$\mathbf{K} = \left(p_{\mathbf{v}}^{\mathcal{F}_c}, p_{\mathbf{v}}^{\boldsymbol{\tau}_c}, p_{\mathbf{v}}^{\mathbf{P}_k}, p^{\mu}\right)_{(k,\mathbf{v})\in\mathcal{F}\times\mathcal{R}_{\mathrm{obj.}}} \tag{15}$$

In particular, attaching $\mathcal{R}_{\mathrm{obj.}}$ to the thumb frame $\mathcal{R}_{\mathrm{th.}} = (\mathbf{t}_0^x, \mathbf{t}_0^y, \mathbf{n}_0)$, with $\mathbf{t}_0^y$ towards the palm, also helps preserve grasp invariances as the four antagonist fingers now share the same coordinate along $\mathbf{n}_0$, reducing $\mathbf{K}$ to 19 elements. Also expressing forces in $\mathcal{R}_{\mathrm{th.}}$ yields a 15-element vector $\mathbf{D}$:

$$\mathbf{D} = (\mathbf{F}_k \cdot \mathbf{v})_{(k,\mathbf{v})\in\mathcal{F}\times\mathcal{R}_{\mathrm{obj.}}} \tag{16}$$

Convertly, finger forces $\mathbf{F}_k$ are directly obtained from $\mathbf{D}$ by projecting $\mathcal{R}_{\mathrm{obj.}}$ coordinates into their local reference frames.

4.3 Neural Network Modelling

Given an object-grasp configuration, we aim at constructing a mapping F between motion and forces such that $\mathbf{D} = F(\mathbf{K})$. In [6], F was modeled with an MLP learning internal forces. Yet, as contact is maintained, forces $\mathbf{D}_i$ at timestep i should depend on the current motion $\mathbf{K}_i$ and the past:

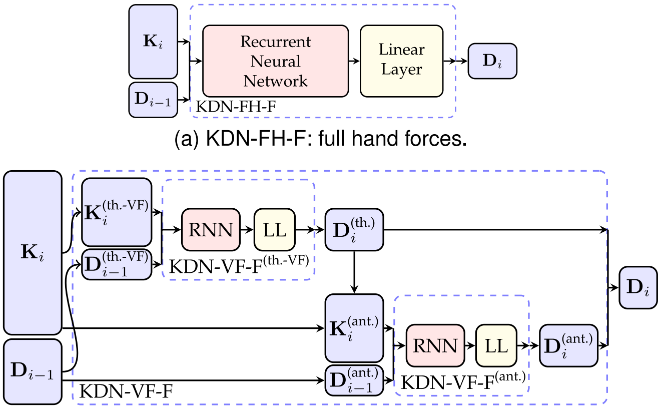
$$\mathbf{D}_i = F\Big((\mathbf{K}_i, \mathbf{D}_{i-1}), \left(\mathbf{K}_j, \mathbf{D}_{j-1}\right)_{j=1,i-1}\Big) \tag{17}$$

This formulation accounts for temporal continuity and helps learn force-motion trajectories rather than single sample associations. We model this sequential structure using recurrent neural networks (RNN) [66] with long short term memory (LSTM) neurons [67], that allow for better learning of long-term dependencies. We investigate four kinodynamics network (KDN) architectures. The first model, KDN-FH-F, predicts full hand forces $\mathbf{D}_i$ from the current kinematics $\mathbf{K}_i$ and previous distribution $\mathbf{D}_{i-1}$ using a single RNN

$$\mathbf{D}_i = \mathrm{KDN-FH-F}(\mathbf{K}_i, \mathbf{D}_{i-1}). \tag{18}$$

Second, we propose a two-stage network, KDN-VF-F, inspired by the virtual finger model. We reduce the four antagonist fingers $\mathcal{F}_{\mathrm{ant.}}$ to a virtual finger associated to their centroid $p_{\mathbf{v}}^{\mathbf{P}_{\mathrm{ant.}}} = <p_{\mathbf{v}}^{\mathbf{P}_k}>_{k\in\mathcal{F}_{\mathrm{ant.}}}$ and average friction coefficient $p^{\mu_{\mathrm{ant.}}} = <\mu_k>_{k\in\mathcal{F}_{\mathrm{ant.}}}$, thus yielding reduced task parameters

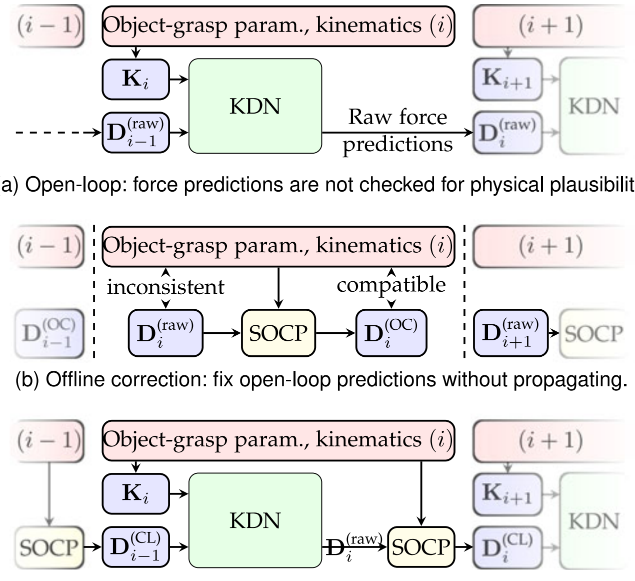
$$\mathbf{K}_i^{(\mathrm{th.-VF})} = \left(p_{\mathbf{v}}^{\mathcal{F}_c}, p_{\mathbf{v}}^{\boldsymbol{\tau}_c}, p_{\mathbf{v}}^{\mathbf{P}_{\mathrm{th.}}}, p^{\mu_{\mathrm{th.}}}, p_{\mathbf{v}}^{\mathbf{P}_{\mathrm{ant.}}}, p^{\mu_{\mathrm{ant.}}}\right)_{\mathbf{v}\in\mathcal{R}_{\mathrm{th.}}}. \tag{19}$$


$\mathbf{K}_i$
Recurrent Neural Network
Linear Layer
$\mathbf{D}_i$
$\mathbf{D}_{i-1}$
KDN-FH-F
$\mathbf{K}_i$
$\mathbf{K}_i^{(\text{th.-VF})}$
$\mathbf{D}_{i-1}^{(\text{th.-VF})}$
RNN
LL
$\mathbf{D}_i^{(\text{th.})}$
KDN-VF-F$^{(\text{th.-VF})}$
$\mathbf{D}_i$
$\mathbf{K}_i^{(\text{ant.})}$
RNN
LL
$\mathbf{D}_i^{(\text{ant.})}$
$\mathbf{D}_{i-1}$
$\mathbf{D}_{i-1}^{(\text{ant.})}$
KDN-VF-F
KDN-VF-F$^{(\text{ant.})}$

(a) KDN-FH-F: full hand forces.

(b) KDN-VF-F: two-stage thumb-virtual finger network.

Fig. 4. Current forces are predicted from current motion and past forces.


$(i-1)$
Object-grasp param., kinematics (i)
$(i+1)$
$\mathbf{K}_i$
KDN
$\mathbf{D}_{i-1}^{(\text{raw})}$
Raw force predictions
$\mathbf{K}_{i+1}$
$\mathbf{D}_i^{(\text{raw})}$
KDN
$(i-1)$
Object-grasp param., kinematics (i)
$(i+1)$
inconsistent
compatible
$\mathbf{D}_{i-1}^{(\text{OC})}$
$\mathbf{D}_i^{(\text{raw})}$
SOCP
$\mathbf{D}_i^{(\text{OC})}$
$\mathbf{D}_{i+1}^{(\text{raw})}$
SOCP
$(i-1)$
Object-grasp param., kinematics (i)
$(i+1)$
$\mathbf{K}_i$
KDN
$\mathbf{K}_{i+1}$
KDN
SOCP
$\mathbf{D}_{i-1}^{(\text{CL})}$
$\mathbf{D}_i^{(\text{raw})}$
SOCP
$\mathbf{D}_i^{(\text{CL})}$

(a) Open-loop: force predictions are not checked for physical plausibility.

(b) Offline correction: fix open-loop predictions without propagating.

(c) Closed-loop: correct force predictions and feed back to the next step.

Fig. 5. Open-loop, offline correction and closed-loop force prediction.

A first RNN then estimates thumb forces $\mathbf{D}_i^{(\text{th.})}$ separately

$$\mathbf{D}_i^{(\text{th.})} = \mathrm{KDN-VF-F}^{(\text{th.}-\text{VF})}\left(\mathbf{K}_i^{(\text{th.}-\text{VF})}, \mathbf{D}_{i-1}^{(\text{th.})}\right). \tag{20}$$

Given thumb force estimates $\mathbf{F}_{\text{th.}}$, force-torque parameters due to antagonist fingers are $p_{\mathbf{v}}^{\mathcal{F}_{\text{ant.}}} = (\mathcal{F}_c - \mathbf{F}_{\text{th.}}) \cdot \mathbf{v}$ and $p_{\mathbf{v}}^{\boldsymbol{\tau}_{\text{ant.}}} = (\boldsymbol{\tau}_c - (\overrightarrow{\mathbf{CP}}_{\text{th.}} \times \mathbf{F}_{\text{th.}})) \cdot \mathbf{v}$, yielding task parameters

$$\mathbf{K}_i^{(\text{ant.})} = \left(p_{\mathbf{v}}^{\mathcal{F}_{\text{ant.}}}, p_{\mathbf{v}}^{\boldsymbol{\tau}_{\text{ant.}}}, p_{\mathbf{v}}^{\mathbf{P}_k}, p^{\mu_{\text{ant.}}}\right)_{(k,\mathbf{v})\in\mathcal{F}_{\text{ant.}}\times\mathcal{R}_{\text{th.}}} \tag{21}$$

A second RNN then learns antagonist finger forces $\mathbf{D}_i^{(\text{ant.})}$

$$\mathbf{D}_i^{(\text{ant.})} = \mathrm{KDN-VF-F}^{(\text{ant.})}\left(\mathbf{K}_i^{(\text{ant.})}, \mathbf{D}_{i-1}^{(\text{ant.})}\right) \tag{22}$$

We depict KDN-FH-F and KDN-VF-F in Fig. 4. Finally, we introduce alternative versions of KDN-FH-F and KDN-VF-F that associate current kinematics $\mathbf{K}_i$ and past forces $\mathbf{D}_{i-1}$ to force variations $\Delta\mathbf{D}_i$. In doing so, we explicitly associate the same output to two sequences that differ by constant internal forces. We denote these alternative architectures by KDN-FH-Δ and KDN-VF-Δ. Full forces are then computed by summing up consecutive predicted force variations.

5 Experiments

We train the four KDN architectures on the manipulation kinodynamics dataset, down-sampled from 400 to 60 Hz for compatibility with vision-based kinematics (Section 6), and split for training (60 percent), validation (20 percent) and testing (20 percent). In KDN-FH-F and KDN-FH-Δ, the RNN contains two hidden-layers of size 256. In KDN-VF-F and KDN-VF-Δ, both RNNs contain a single hidden-layer of size 256. We implement and train the networks within the Torch7 framework [68] using stochastic gradient descent with a mean square error criterion and dropout [69] to avoid overfitting.

5.1 Force Reconstruction Model

As physics is not explicitly enforced by the RNNs, raw force predictions can be inconsistent with the observed motion. We depict such open-loop predictions in Fig. 5a. Using these as target forces $\widetilde{\mathbf{F}}_k$ in the SOCP of Fig. 3 allows the extraction of physically plausible forces in their vicinity. This can be done after collecting a complete sequence of open-loop predictions. We depict this offline correction process in Fig. 5b. However, as the RNNs are trained on physically consistent data, repeatedly feeding inconsistent motion-force sequences may lead to growing errors. Force predictions can be corrected between time steps by integrating the SOCP in closed-loop with the KDN, depicted in Fig. 5c.

We compute the estimation errors (average and standard deviation) for the four network architectures using open-loop prediction, offline correction or closed-loop prediction, on complete manipulation sequences with $\mathbf{D}_0$ initialized from ground-truth forces in Eq. (17). We report the results in Table 1, highlighting for each neural network architecture the lowest mean and standard deviation errors across correction processes. Humans being able to apply infinitely many force distributions to produce the same motion results in some intrinsic variability even in performing the same task repeatedly. Indicatively, standard deviations values between 0 and 2N were reported on static finger pressing tasks in [70]. We argue that this intrinsic variability is likely greater in the case of unconstrained, dynamic motions, though difficult to measure confidently.

We observe from Table 1 that offline correction and closed-loop prediction outperform open-loop prediction on all architectures. This is especially the case for the networks estimating force variations $\Delta\mathbf{D}_i$, as these tend to be rather unstable and prone to drift. For instance, in Fig. 6, the open-loop predictions rapidly drift away from the net force and torque producing the target kinematics. Additionally, the individual normal forces become negative, which would

TABLE 1
Force Estimation Errors on Complete Sequences-Mean (Std. Dev.) [N]

	Open-loop	Offline corr.	Closed-loop
KDN-FH-F	0.49 (4.14)	0.44 (4.07)	**0.16** (**3.54**)
KDN-FH-Δ	−43.67 (156.72)	0.60 (**4.74**)	**0.50** (11.03)
KDN-VF-F	0.29 (3.19)	0.29 (3.13)	**0.12** (**2.60**)
KDN-VF-Δ	1145.06 (3984.86)	3.54 (11.80)	**2.32** (**6.60**)

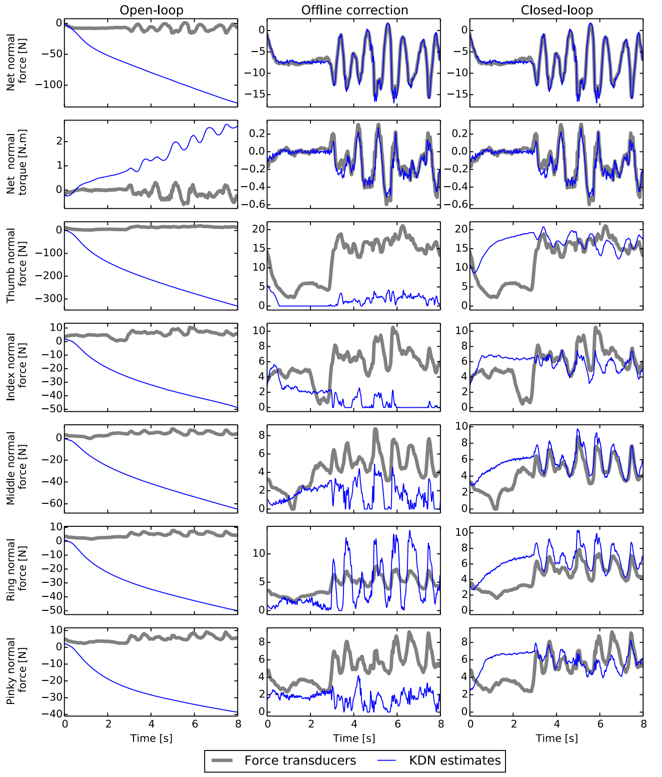


Fig. 6. Open-loop, offline-corrected and closed-loop force predictions for KDN-VF-Δ. Open-loop forces drift away from physically plausible forces, becoming negative. Compatibility with the observed motion is enforced through offline correction or closed-loop control at each time step.

mean that fingertips pull rather than press on the contact surface. Offline correction searches for physically valid forces in the vicinity of negative raw predictions, finally yielding distributions of minimal norm. In contrast, closed-loop prediction can help the network recover from incorrect predictions and maintain human-like grasping forces. Overall, the networks predicting force distributions generally perform better than those estimating force variations. For the former, offline correction does not appear to significantly improve the open-loop estimations, which shows that these RNNs are rather successful at capturing the relationship between kinematics and underlying forces. Finally, the better accuracy of KDN-VF-F indicates that the virtual finger model can be a useful tool to decouple the static indeterminacy stemming from the thumb and antagonist fingers. Still, the two-stage architecture makes KDN-VF-Δ more prone to drift since thumb force predictions cannot be corrected alone before computing the antagonist forces.

5.2 Force Drift Over Time

Due to the infinity of force distributions compatible with a given motion, the force predictions are likely to deviate from the transducer measurements over time. We quantify this effect by splitting the experiments into sub-sequences of maximum duration 1, 2, 4, 8, 16, 32 s and computing the resulting estimation errors for the four architectures with ground-truth initialization and either offline correction or closed-loop prediction. For completeness, we reproduce the force estimation errors over the full length sequences (average duration 60.1 s, standard deviation 3.8 s). We report the results in Table 2, highlighting for each duration the lowest errors across the proposed architectures.

In line with the observations made on the full-length experiments, KDN-VF-Δ is the worst-performing network for every sequence duration, whereas KDN-VF-F is consistently best-performing in terms of standard deviation. This indicates again that decoupling thumb and antagonist redundancies is a viable strategy, yet more unstable in the presence of force variation uncertainties. We also observe that KDN-FH-F outperforms KDN-VF-F on sequence durations 4, 8, 16 and 32 s in terms of average error. These results are representative of two partially competing goals in force estimation. On one hand, estimating the full hand forces together, the KDN-FH-F architecture tends to predict an average plausible distribution based on past motion-forces associations. On the other hand, the separate estimation of the thumb force in the first stage of the virtual finger architecture explicitly decouples the major indeterminacy between thumb and antagonist fingers, enabling KDN-VF-F to better track the individual variability across subjects. We also observe that KDN-FH-Δ yields better results than its full force counterpart KDN-FH-F on the 1 s sequence duration and 2 s to a lesser extent. Recall that the $\Delta\mathbf{D}_i$ networks were introduced to accommodate the possibility of having the same motion caused by an infinity of force distributions. It appears here that KDN-FH-Δ is better at matching the real force variations on short sequences. Still, the applicability of this result on real manipulation tasks is hindered by, first, the accumulation of $\Delta\mathbf{D}_i$ prediction errors over time, and second, the accuracy of the predicted force sequence being contingent on its initialization from real force measurements.

5.3 Force Sequence Initialization

Manipulation forces are sequentially computed based on an initial distribution that can be adjusted freely. We assess the force variability following non ground-truth initialization for sequences of maximum duration 4.0, 8.0, 16.0 and 32.0 s.

TABLE 2
Force Estimation Errors for Proposed Prediction-Correction Architectures over Increasing Sequence Durations-Mean (Std. Dev.) [N]

Sequence duration	1.0 s	2.0 s	4.0 s	8.0 s	16.0 s	32.0 s	Full length
KDN-FH-F, offline correction	−0.21 (2.06)	−0.21 (2.43)	−0.13 (2.86)	−0.04 (3.22)	**0.07** (3.54)	0.19 (3.76)	0.44 (4.07)
KDN-FH-F, closed-loop	−0.13 (2.20)	−0.12 (2.47)	−0.07 (2.80)	**0.00** (3.07)	0.06 (3.24)	0.08 (3.33)	0.16 (3.54)
KDN-FH-Δ, offline correction	**0.00** (**1.80**)	0.15 (2.42)	0.36 (3.22)	0.56 (3.89)	0.68 (4.34)	0.56 (4.62)	0.60 (4.74)
KDN-FH-Δ, closed-loop	0.02 (1.87)	0.11 (2.48)	0.27 (3.44)	0.45 (5.14)	0.58 (7.39)	0.57 (9.32)	0.50 (11.03)
KDN-VF-F, offline correction	0.07 (2.09)	0.13 (2.51)	0.20 (2.82)	0.25 (2.99)	0.27 (3.07)	0.28 (3.11)	0.29 (3.13)
KDN-VF-F, closed-loop	0.02 (1.86)	**0.04** (**2.16**)	**0.07** (**2.38**)	0.10 (**2.50**)	0.11 (**2.56**)	**0.12** (**2.58**)	**0.12** (**2.60**)
KDN-VF-Δ, offline correction	0.43 (2.93)	0.87 (4.47)	1.64 (7.11)	2.37 (9.33)	2.90 (10.61)	2.94 (11.13)	3.54 (11.80)
KDN-VF-Δ, closed-loop	0.41 (2.47)	0.76 (3.45)	1.24 (4.74)	1.69 (5.69)	1.99 (6.17)	2.15 (6.43)	2.32 (6.60)

TABLE 3
Force Estimation Errors for Ground-Truth versus Random Force Initialization Over Increasing Sequence Durations-Mean (Std. Dev.) [N]

Duration	4.0 s		8.0 s		16.0 s		32.0 s	
Initialization	Reference	Random	Reference	Random	Reference	Random	Reference	Random
KDN-FH-F, OC	−0.13 (**2.86**)	−**0.00** (3.42)	−**0.04** (**3.22**)	0.12 (3.60)	**0.07** (**3.54**)	0.21 (3.76)	**0.19** (**3.76**)	**0.19** (3.80)
KDN-FH-F, CL	−**0.07** (**2.80**)	0.09 (3.36)	**0.00** (**3.07**)	0.10 (3.43)	**0.06** (**3.24**)	0.09 (3.42)	0.08 (**3.33**)	**0.06** (3.36)
KDN-FH-Δ, OC	0.36 (**3.22**)	**0.34** (3.72)	**0.56** (**3.89**)	0.52 (4.25)	0.68 (**4.34**)	**0.64** (4.49)	0.56 (**4.62**)	**0.52** (4.73)
KDN-FH-Δ, CL	**0.27** (**3.44**)	0.37 (4.08)	**0.45** (**5.14**)	0.53 (5.75)	**0.58** (7.39)	0.63 (**7.35**)	0.57 (**9.32**)	**0.56** (9.59)
KDN-VF-F, OC	**0.20** (**2.82**)	0.22 (3.01)	**0.25** (**2.99**)	0.27 (3.08)	**0.27** (**3.07**)	0.28 (3.13)	**0.28** (**3.11**)	0.29 (3.14)
KDN-VF-F, CL	**0.07** (**2.38**)	0.12 (2.61)	**0.10** (**2.50**)	0.12 (2.63)	**0.11** (**2.56**)	0.13 (2.63)	**0.12** (**2.58**)	0.13 (2.63)
KDN-VF-Δ, OC	**1.64** (**7.11**)	1.79 (7.55)	**2.37** (**9.33**)	**2.37** (9.50)	2.90 (10.61)	**2.70** (**10.32**)	**2.94** (11.13)	2.99 (**11.10**)
KDN-VF-Δ, CL	**1.24** (**4.74**)	1.27 (5.11)	**1.69** (**5.69**)	1.75 (5.86)	**1.99** (**6.17**)	2.06 (6.29)	**2.15** (**6.43**)	2.18 (6.47)

Each sequence is initialized as follows. Using the average and standard deviation $\boldsymbol{\mu}, \boldsymbol{\sigma}$ of each finger force throughout the manipulation kinodynamics dataset, we pick a random sample $\widetilde{\mathbf{D}}_0$ following the normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. We then correct $\widetilde{\mathbf{D}}_0$ using the SOCP of Section 4.1. Thus, we ensure that the resulting distribution $\mathbf{D}_0$ is compatible with the initial kinematics $\mathbf{K}_0$. We report the force estimation errors for random and ground-truth initialization in Table 3, highlighting, for each duration and architecture, the lowest errors between the two initialization methods.

Expectedly, ground-truth initialization yields better force estimates overall. Still, for each architecture, the performance difference decreases with the sequence duration. Indeed, even when starting from the same distribution, the predicted sequence is likely to deviate from the transducer measurements due to the infinity of force variations producing the same motion. This mitigates the importance of the force initialization over time. In the case of the best-performing network, KDN-VF-F (closed-loop), the difference is actually minor even starting from 8.0 s sequences. Finally, note that for any initial force distribution, the resulting sequence is constructed to be physically plausible given the observed motion and compatible with the forces a human could likely apply, based on the manipulation kinodynamics dataset. This allows the generation of force sequences following different profiles for the same motion (e.g., light or strong starting grasp). This method can also be used to reinitialize the prediction model when the resulting distributions are unreliable, as it may happen in the presence of motion tracking discontinuities.

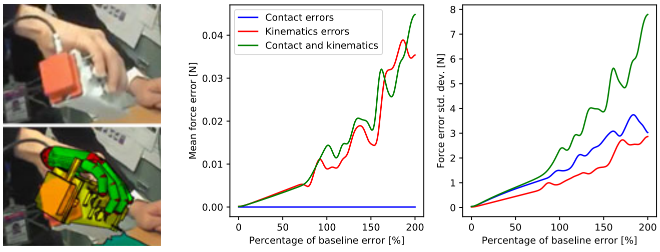
6 Force Sensing from Vision

We propose to estimate finger forces from markerless visual tracking, circumventing any instrumentation whatsoever.

6.1 Model-Based Tracking

Force estimation requires the kinematics of the object and contact point locations. Due to the intensity of the interaction (firm grasps, high velocities) and its observability (occlusions, motion blur), the 3D state of the object cannot be computed independently of that of the hand. The requirement for joint tracking in such situations is discussed in [31], [32], [71]. We thus tackled hand-object tracking using the approach of [71]. A method that automatically computes contact points for force estimation was presented in [6]. This method did not apply well in our case due to the larger variety of motions (high velocities) and objects (easily occluded). In our preliminary experiments, tracking the hand together with the object produced both inaccurate object poses and contact locations (in the order of 20 mm), see Fig. 7a. To assess the effect of contact point errors, we decomposed force transducer measurements on AHRS kinematics (see Fig. 3), using either ground-truth contact points or perturbed by a Gaussian noise of standard deviation 0 to 20 mm. We depict the resulting force reconstruction errors in Fig. 7b. Net forces are only determined by the motion, hence zero mean errors. However, different contact point enable different ways of distributing the same net forces, hence growing force variability. Since the best performing KDN architecture achieves a lowest standard deviation error of about 2.60 N, we consider it as an upper bound on acceptable force reconstruction errors due to kinematics. Beyond that (possibly even before, due to uncertainties adding up), contact point errors are guaranteed to counterbalance the accuracy of the force predictions. 20 mm contact point errors yield about 3 N force reconstruction error (std. dev.), and thus cannot be used for accurate force prediction.

In our work, all recorded motions and forces are annotated with ground-truth contact locations (i.e., that of the force transducers), which we use to circumvent their estimation. We implement the hypothesis that the grasp pose does not change noticeably during manipulation and thus consider the object and the hand as a single, rigid compound. This reduces the problem of tracking an articulated hand and a rigid object (34D) to only a rigid object (7D), significantly increasing efficiency and robustness under



(a) Hand-object tracking failure.

(b) Propagation of contact point and kinematics estimation errors to force reconstruction.

Fig. 7. Tracking the articulated hand pose together with the object rigid motion produces (a) inaccurate object and contact positions that (b) propagate to forces. Baseline errors (std. dev.): contact: $10\ \mathrm{mm}$ (pos.), kinematics: $2.9\ \mathrm{ms}^{-2}$ (trans. acc.), $30.9\ \mathrm{rad\ s}^{-1}$ (rot. vel.) .

mutual occlusions. The origin of the rigid compound in object space is set to the origin of the object alone in object space. This makes the rigid compound's trajectory that of the object itself. To implement this hypothesis in 3D tracking, we 1) track both hand and object during the entire sequence using [71], 2) select a frame to form the rigid compound and 3) track the rigid compound only using [71].

Step 1) is fully automatic. We use [71] to generate an initial estimate of the trajectories for the hand and the object. Despite noise, fast motion and the lack of any problem-specific prior, the result is visually plausible but the truth that fingers touch the object at the known contact points is not reconstructed accurately enough. To have the method generate results of higher fidelity, a large computational budget is given to the optimizer (particle swarm optimization, PSO [72]), namely 100 particles and 100 generations per entity (hand, object). The GPU-parallelized implementation of [71] yields a processing rate of 0.5 fps.

Step 2) is partially automatic. Its goal is to yield a hand-object compound that best matches the actual grasp realized by the subject. To this end, from the entire sequence, a single frame is sought whose back-projection maximizes a) the image area corresponding to the hand, b) the viewable surface percentage of the hand's 3D mesh, c) the image area corresponding to the object, d) the viewable surface percentage of the object's 3D mesh and e) in which the hand grasps the object. Though we performed this selection empirically, we believe it can clearly be automated by quantifying each of the aforementioned criteria and formulating an objective function to be optimized over frame indices. Currently, due to manual intervention, this step takes the most time. For the selected frame, a single tracking step of [71] is executed anew, but with an additional term that favors hand-object hypotheses where the fingertips touch the known contact points. More specifically, let $E(h, o)$ be the error function used in [71], for a hand pose h and an object pose o. Then, a new objective function $E'(h, o)$ is defined as

$$E'(h, o) = E(h, o) + \lambda \sum_{k \in \mathcal{F}} \|S_k(h) - T_k(o)\|, \tag{23}$$

with $\mathcal{F}$ the set of fingers, $S_k(h)$ the world space position of fingertip k under hand pose hypothesis h, and $T_k(o)$ the world space position of the known contact point for finger k under object pose hypothesis o. This new objective is minimized using the optimization scheme of [71] to yield a new hand-object configuration which is a balance between i) matching the observations, as in [71] and ii) making sure the finger tips touch the contact points. Search for the hand-object poses is initialized from the tracking results in the preliminary tracking of Step 1). After optimal hand and object poses are computed, a new geometry is introduced, that is the union of the hand and object geometries, under those new poses. This new geometry models the object to be further tracked. This step is computationally similar to Step 2) and only slightly increased in order to compute the additional term. In our experiments we used $\lambda = 0.0002$.

Step 3) is fully automatic. It makes use of the tracking result of Step 1) and the geometry of Step 2) to initialize and track a single rigid compound across the same sequence, again employing [71]. Its result is the trajectory of the compound, which also constitutes the trajectory of the object alone. Computationally, this is the lightest tracking step and can be performed at processing rates higher than 30 fps.

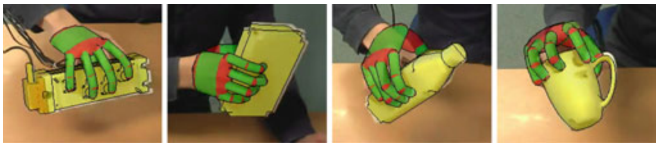


Fig. 8. The hand and the object are tracked as a rigid compound.

6.2 Kinematics Estimation from Visual Tracking

With the camera calibrated intrinsically and extrinsically such that the vertical direction in the world frame is known (to account for gravitation **g**, see Section 3.3), we record and process 12 tracking experiments using the following objects: First, the instrumented device of Section 3, in a configuration that does not appear in the training dataset (mass 279 g). Second, three objects used in daily activities, 3D-printed and equipped with AHRS and force transducers for ground truth: a cuboid box (856 g), a small bottle (453 g), and a mug (174 g). We use the latter as an application of the force model on non-prismatic grasps in Section 7.2. We depict sample tracking results in Fig. 8.

Given the pose of the object throughout the experiment, we estimate its first and second-order kinematics (i.e., velocity and acceleration) by numerical differentiation. Due to noise in the estimated trajectory, direct differentiation of the tracked pose signal (e.g., by central difference) results in large spikes in the derivatives, and thus the estimated forces. This effect can be mitigated by smoothing the original signal before differentiation. However, force profiles occurring in manipulation tasks are in fact spiky (see Fig. 6). Thus, smoothing a trajectory comes at the risk of suppressing not only noise, but actual acceleration spikes.

Besides direct differentiation by central difference, we evaluate two smoothing techniques on visual tracking trajectories: Gaussian filtering and algebraic numerical differentiation [73]. In the former, the initial signal is smoothed by convolution with a Gaussian function $G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2})$, with standard deviation σ. Conveniently, differentiating the smoothed signal also amounts to convolving the original signal with the derivatives of G. Denoting by $T_s = 1/60$ s the sampling period, we experimentally choose $\sigma = 3T_s$ and truncate G at $\pm 4\sigma$. Alternatively, we investigate the use of algebraic numerical differentiators [73], which do not depend on the noise statistical properties and previously produced good kinematics estimates in [6]. Such filters are characterized by parameters (κ, μ), akin to the Gaussian σ, and a temporal window of half-width T. We empirically choose $T = 4T_s$ and $\kappa = \mu = 0.5$. For each differentiation method, we compare the resulting translational accelerations and rotational velocities to those measured by the AHRS. Indicatively, we quantify the effect of kinematics estimation errors on force distributions by computing distributions that are closest to force transducer measurements as in Fig. 3, but using the kinematics estimates provided by each method. We report the resulting errors in Table 4, highlighting, for each motion and force quantity, the lowest error across differentiation methods.

TABLE 4
Kinematics and Force from Vision-Mean (Std. Dev.) Errors From Central Difference, Gaussian Filtering and Algebraic Differentiation

	Central	Gaussian	Algebraic
Trans. acc.$[\mathrm{m\ s^{-2}}]$	$0.31(25.36)$	$-\mathbf{0.02}(\mathbf{2.92})$	$-0.05(3.03)$
Rot. vel. $[\mathrm{rad\ s^{-1}}]$	$0.14(446.45)$	$-0.05(\mathbf{30.94})$	$\mathbf{0.01}(31.76)$
Force $[\mathrm{N}]$	$1.18(8.94)$	$\mathbf{0.01}(\mathbf{0.72})$	$\mathbf{0.01}(0.75)$

On typical tracking sequences, smoothing techniques appear necessary to compute reliable kinematics estimates. Both Gaussian and algebraic filters yield reasonable force discrepancies despite possible tracking uncertainties and discontinuities. Overall, while the Gaussian filter seems to perform slightly better than the algebraic filter, the latter also requires significantly less samples per estimate. This allows for a shorter lag for real-time applications while also better capturing high frequency force variations, at the cost of a slightly larger sensitivity to tracking noise. To further quantify the effects of tracking errors on contact points, we also compute force reconstruction errors for kinematics estimates perturbed by between 0 and 200 percent of the errors reported in Table 4 for Gaussian filtering, depicted in Fig. 7b. In contrast with contact points, kinematics errors directly influence the net force, hence non-zero mean errors, that also grow in standard deviation. Interestingly, adding contact errors has no additional effect on the mean force error as explained in Section 6.1, but having both kinematics and contact errors leads to larger error standard deviations than either individually. This graph is also informative of the level of accuracy that is needed to reach a given precision. For example, the upper bound in admissible force estimation errors of 2.60 N discussed in Section 6.1 is reached at 200 percent of the current error baseline for rigid compound tracking (kinematics errors only, red plot). Tracking errors beyond that should systematically be rejected for the purpose of force estimation. Also, the same graph suggests that to reach the 1N precision that the rigid compound tracking currently achieves (red plot at 100 percent baseline errors), yet without enforcing a static grasp (contact and kinematics errors, green plot), contact errors must be below 5mm and tracking must become twice as accurate (50 percent baseline errors).

6.3 Force Prediction from Vision-Based Kinematics

Using a single camera, we track manipulation experiments and estimate the object's kinematics with algebraic filtering. In Section 5, although the four network architectures are trained on AHRS data, the object's kinematics is used as an input without consideration of the way it is measured. Thus, the trained networks can seamlessly generate force sequences from vision-based kinematics. In order to be completely independent of ground-truth sensing, we use the random initialization process described in Section 5.3. We compute the resulting estimation errors with respect to ground-truth force transducer measurements, along with, for reference, force predictions derived from the AHRS kinematics; none of these being used in the vision-based estimation process. We report our results in Table 5, highlighting, for each motion type and initialization method, the lowest errors across prediction-correction architectures.

TABLE 5
Force Estimation Errors From AHRS versus Vision - Mean (Std. Dev.) [N]

Kinematics Initialization	AHRS ground truth	AHRS random	Vision random
KDN-FH-F, OC	-1.10 (2.95)	-1.12 (2.95)	-1.18 (3.11)
KDN-FH-F, CL	-1.37 (3.12)	-1.37 (3.13)	-1.25 (3.61)
KDN-FH-Δ, OC	0.72 (3.38)	0.85 (3.42)	0.94 (3.39)
KDN-FH-Δ, CL	1.21 (5.80)	2.27 (11.86)	3.50 (17.28)
KDN-VF-F, OC	0.18 (2.64)	0.14 (2.68)	0.15 (2.69)
KDN-VF-F, CL	$-\mathbf{0.01}$ (**2.20**)	**0.02** (**2.27**)	$-\mathbf{0.04}$ (**2.30**)
KDN-VF-Δ, OC	5.40 (27.61)	5.16 (23.06)	5.94 (24.54)
KDN-VF-Δ, CL	2.20 (16.31)	3.87 (19.99)	7.37 (25.15)

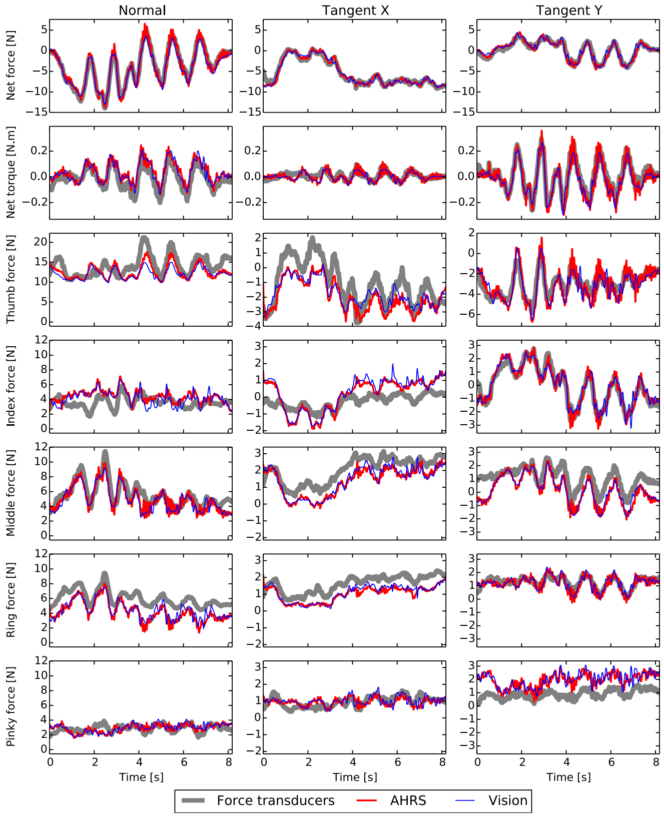


Fig. 9. Forces from closed-loop KDN-VF-F and random initialization.

Under the same initialization conditions, forces computed from vision are comparable to forces computed from AHRS measurements. The decrease in accuracy is most noticeable on networks estimating force variations $\Delta\mathbf{D}_i$ due to a higher tendency to drift, as discussed in Section 5, but also additional uncertainties from visual tracking. We depict an example of forces estimated from vision in Fig. 9. Tracking discontinuities (e.g., lost hand-object pose), following second-order differentiation, are perceived by the force estimation framework as acceleration spikes and result in sudden fingertip force variations. These errors accumulate in the case of $\Delta\mathbf{D}_i$ networks since each prediction is directly relative to the preceding sample. When erroneous kinematics can be identified, their impact can be mitigated by reinitializing the prediction process based on the last reliable sample. However, while doing so is straightforward when AHRS measurements are available, it is difficult from the tracked kinematics alone, since acceleration spikes are not

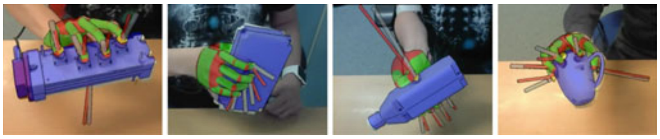


Fig. 10. Forces predicted from vision (red) versus ground-truth (grey).

necessarily due to discontinuities but can also stem from actual sudden motions. Overall, KDN-VF-F appears the most resilient architecture to visual tracking uncertainties. We depict sample force prediction results in Fig. 10.

7 Discussion

7.1 Visual Tracking Assumptions

In Section 6.1, we suppose the contact points known and use them to compute a static grasp throughout the motion. Note that our force estimation framework itself is independent of the tracking method employed as long as reliable motion and contact information can be provided. The difficulty for us was to collect ground-truth measurements to validate our approach. Therefore, we forced the positioning of the fingertips at desired locations for both the real objects and the visual tracking system. Indeed, to allow arbitrary finger placement, the experimental apparatus should be covered with an array of high-precision 3D force transducers (that are not available in the required dimensions), or alternatively with dedicated force sensing surfaces [74], generally limited in accuracy and range (e.g., normal forces only).

Our force estimation framework can readily challenge in-hand manipulation scenarios with more sophisticated tracking systems (e.g., multi-camera). Again, assessing such tasks is limited by the difficulty of measuring the actual forces without obstructing the subject's haptic sense, which we consider essential in our demonstration. In effect, the tracking method we describe does not introduce any constraint besides those relative to the ground-truth instrumentation, while making it possible to monitor manipulation forces using a single off-the-shelf depth sensor.

7.2 Beyond Prismatic Grasps

We evaluate the force estimation framework on a non-prismatic grasp using a mug-shaped device (see Fig. 1f). Force transducers are arranged on a circle, with the contact normals pointing towards the center. We compute force distributions using either visual tracking or AHRS and depict the resulting predictions in Fig. 11. First, we observe that by considering the hand and the object as a single rigid compound, we are able to track the mug fairly accurately using a single depth sensor, despite it being essentially rotationally symmetric, except for a handle that is easily occluded. Second, RNN predictions do not follow the subtle force variations along the normal $\mathbf{n}_k$ and tangential directions $\mathbf{t}_k^x$ as closely as the tangential direction $\mathbf{t}_k^y$. Recall from Section 4.2 that the individual $\mathbf{t}_k^y$ per finger are defined, uniformly, as oriented towards the palm. This property is preserved in the case of the mug. However, while for prismatic grasps the $\mathbf{n}_k$ are collinear with each other and perpendicular to the $\mathbf{t}_k^x$, couplings appear between and among each set in the case of the mug. Still, the SOCP ensures that the final distributions are physically plausible based solely on the observed kinematics and the object-grasp properties, regardless of the RNN training dataset. We also observe that predicted and measured thumb forces remain rather comparable. This suggests that knowledge acquired from prismatic grasps on the 3-dimensional indeterminacy between thumb and antagonist fingers can generalize and serve as a prior during physics-based optimization (e.g., to compute the thumb force separately and redistribute the complementary antagonist force with a weighted L^2 criterion). Such a prior can also be used to predict forces on grasps involving less than five fingers, with additional SOCP equality constraints ensuring that non-contacting fingers do not exert forces.

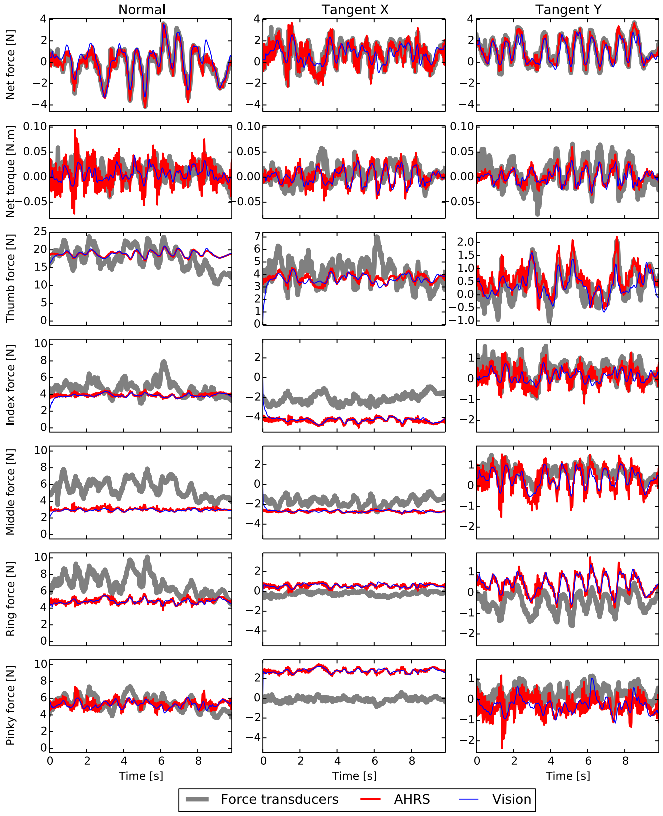


Fig. 11. Force estimates with non-prismatic grasp (mug).

While we could imagine extending the force estimation framework further by training new network architectures on arbitrary grasps, this is difficult in practice. The current ground-truth instrumentation captures 11 contact space degrees of freedom (grasp width and 2D tangential finger positions). In contrast, for general grasps, the instrumentation should allow 25 degrees of freedom (5 per finger, ignoring transducer orientations about the normal axes). Due to the greater dimensionality, it would require significantly more experiments to obtain a diverse and extensive dataset, as well as a much heavier experimental setup to fine-tune the position and roll-pitch of each transducer independently.

7.3 Computational Performance

On a computer equipped with an Intel i7-4700MQ CPU (quad-core 2.40 GHz) and an NVIDIA GTX 780M GPU, we apply the KDN-VF-F closed-loop architecture on the testing dataset (39 experiments, total duration 2470 s, 60 Hz). We report computation times in Table 6. While the total computation time appears greater than the dataset duration, the

TABLE 6
Computation Time Decomposition by Process

	Total	Per sample	Per timestep
Experiment duration	2470.0 s	16.67 ms	100.00%
Computation time	3521.4 s	23.76 ms	142.57%
Algebraic diff.	22.3 s	0.15 ms	0.90%
RNN prediction	120.4 s	0.81 ms	4.87%
$\hookrightarrow$ Data formatting	86.2 s	0.58 ms	3.49%
SOCP correction	641.8 s	4.33 ms	25.98%
$\hookrightarrow$ Initialization	659.0 s	4.45 ms	26.68%
Lua/Python bridge	1991.7 s	13.44 ms	80.64%

decomposition per process shows that the current implementation is actually rather sub-optimal and the three core components take only 5.29 ms per sample. First, algebraic differentiators implemented as finite impulse response filters are of minor impact on computation times. Second, RNN predictions are parallelized on the GPU using the Torch7 framework [68]. Third, SOCP solving is done with the CVXOPT library [75]. A typical iteration is as follows:

(1) Given current kinematics and previous corrected forces $\mathbf{F}_{i-1}$, we construct the RNN input vector $(\mathbf{K}_i, \mathbf{D}_{i-1})$.
(2) The network produces a raw force prediction $\mathbf{D}_i^{(\mathrm{raw})}$.
(3) We assemble SOCP constraint matrices from the target kinematics, and the cost function from $\mathbf{D}_i^{(\mathrm{raw})}$.
(4) We solve the SOCP and get the corrected forces $\mathbf{F}_i$.

Steps 1 and 2 are executed in Lua for Torch7, while steps 3 and 4 are executed in Python for CVXOPT. Both being interpreted languages explains part of the overhead in preparing the data for each process. However, the majority of the processing time is actually spent on managing the two interpreters in succession without actually performing calculation (Lua/Python bridge value in Table 6). Thus, simply implementing our method within a unified computational framework would certainly yield a tremendous increase in performance enabling real-time use. Other possible computational improvements include refactoring data structures to reduce redundancies and update constraint matrices only when needed, initializing the SOCP search at the RNN predictions, and rewriting the physical plausibility problem as a quadratic program (QP) with a discretized friction cone.

8 Conclusion and Future Work

Our work establishes that monitoring hand-object interaction forces at the fingertip level can be addressed in a cheap, reliable and transparent way using vision. Based on the first large-scale dataset on manipulation kinodynamics, our approach estimates force distributions that are compatible with both physics and real human grasping patterns. While the case of static prismatic grasps may appear restrictive, this limitation is only relative to the instrumentation required to collect ground-truth measurements. Provided an extended experimental setup, we expect that our method can seamlessly extend to arbitrary grasps. Besides, the current SOCP formulation is independent of the training dataset and always produces forces causing the observed motion. Finally, even limited to prismatic grasps, the estimation of 3D forces for all five fingers on arbitrary motions greatly extends the state of the art in interaction capture. Our approach is achieved with a single RGB-D camera, which enables its use for monitoring of human activities and robot learning from demonstration in daily settings.

Our approach is readily compatible with any tracking method providing accurate object kinematics. We present such qualitative results [76], [77] in the supplementary material,[2] using hand-picked contact locations. To monitor non rigid grasps, we aim to apply the force estimation framework in conjunction with tracking to guide the pose search as an implicit model for grasp plausibility and realism [78]. Our work would also benefit from future advances in hand tracking and force transducing technologies. The former would facilitate data acquisition on diverse objects for learning on visual data only. Individual finger movements (e.g., contact sliding or fingertip deformations) could yield subtle clues on the subject's perceived effort and be informative of even slight changes in manipulation forces. Our future work involves alleviating current instrumentation limitations and implementing soft finger contact models [64] for dextrous manipulation. The generalization to arbitrary grasps could be addressed by considering the variability of manipulation forces as an inverse optimal control problem involving physiological criteria, e.g., grasp efficiency [79]. It would also be valuable to find a systematic way to quantify visual tracking accuracy requirements to achieve a target force estimation precision. In the long term, we plan to extend the force estimation framework to general articulated bodies for bi-manual grasping and whole-body interactions [80].

Acknowledgments

This work was partially supported by the FP7 EU RoboHow.Cog project (www.robohow.eu), and the Japan Society for Promotion of Science (JSPS) Kakenhi B No. 16H02886.

References

[1] A. Gupta, A. Kembhavi, and L. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1775–1789, Oct. 2009.

[2] Y. Zhu, Y. Zhao, and S. Chun Zhu, "Understanding tools: Task-oriented object modeling, learning and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2855–2864.

[3] C. Ye, Y. Yang, C. Fermuller, and Y. Aloimonos, "What can I do around here? deep functional scene understanding for cognitive robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4604–4611.

[4] X. Niu, A. V. Terekhov, M. L. Latash, and V. M. Zatsiorsky, "Reconstruction of the unknown optimization cost functions from experimental recordings during static multi-finger prehension," *Motor Control*, vol. 16, no. 2, pp. 195–228, 2012.

[5] G. Slota, M. Latash, and V. Zatsiorsky, "Grip forces during object manipulation: experiment, mathematical model, and validation," *Exp. Brain Res.*, vol. 213, no. 1, pp. 125–139, 2011.

[6] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2810–2819.

[7] S. A. Mascaro and H. H. Asada, "Photoplethysmograph fingernail sensors for measuring finger forces without haptic obstruction," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 698–708, Oct. 2001.

[8] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Estimation of fingertip force direction with computer vision," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1356–1369, Dec. 2009.

[2] https://www.youtube.com/watch?v=NhNV3tCcbd0

[9] S. Urban, J. Bayer, C. Osendorfer, G. Westling, B. B. Edin, and P. van der Smagt, "Computing grip force and torque from finger nail images using Gaussian processes," in *Proc. IEEE-RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 4034–4039.

[10] P. G. Kry and D. K. Pai, "Interaction capture and synthesis," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 872–880, 2006.

[11] J. M. Rehg and T. Kanade, "Visual tracking of high dof articulated structures: An application to human hand tracking," in *Proc. Eur. Conf. Comput. Vis.*, 1994, pp. 35–46.

[12] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 101.1–101.11.

[13] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3D hand pose estimation from monocular video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1793–1805, Sep. 2011.

[14] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 852–863.

[15] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1106–1113.

[16] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3786–3793.

[17] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, 2014, Art. no. 169.

[18] P. Krejov, A. Gilbert, and R. Bowden, "Combining discriminative and model based approaches for hand pose estimation," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2015, pp. 1–7.

[19] G. Rogez, J. S. Supancic, and D. Ramanan, "Understanding everyday hands in action from RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3889–3897.

[20] M. Cai, K. M. Kitani, and Y. Sato, "A scalable approach for understanding the visual structures of hand grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1360–1366.

[21] D.-A. Huang, M. Ma, W.-C. Ma, and K. M. Kitani, "How do we use our hands? discovering a diverse set of common grasps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 666–675.

[22] T. Sharp, et al., "Accurate, robust, and flexible real-time hand tracking," in *Proc. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 3633–3642.

[23] J. Taylor, et al., "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 143.

[24] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1868–1876.

[25] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3793–3802.

[26] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNS," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3593–3601.

[27] A. Sinha, C. Choi, and K. Ramani, "Deephand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4150–4158.

[28] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 680–689.

[29] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3D convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1991–2000.

[30] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[31] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2088–2095.

[32] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1862–1869.

[33] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing hands in action using discriminative salient points and physics simulation," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 172–193, 2016.

[34] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from RGB-D input," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 294–310.

[35] N. Kyriazis and A. Argyros, "Physically plausible 3D scene tracking: The single actor hypothesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 9–16.

[36] W. Zhao, J. Zhang, J. Min, and J. Chai, "Robust realtime physics-based motion control for human grasping," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 207:1–207:12, 2013.

[37] Y. Wang, et al., "Video-based hand manipulation capture through composite motion control," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 43.

[38] M. A. Arbib, T. Iberall, and D. Lyons, "Coordinated control programs for movements of the hand," *Hand Function Neocortex*, vol. 10, pp. 111–129, 1985.

[39] F. Gao, M. L. Latash, and V. M. Zatsiorsky, "Internal forces during object manipulation," *Exp. Brain Res.*, vol. 165, no. 1, pp. 69–83, 2005.

[40] J. Kerr and B. Roth, "Analysis of multifingered hands," *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 3–17, 1986.

[41] T. Yoshikawa and K. Nagai, "Manipulating and grasping forces in manipulation by multifingered robot hands," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 67–77, Feb. 1991.

[42] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA, USA: MIT Press, 1985.

[43] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, 1994.

[44] J. R. Flanagan and R. S. Johansson, "Hand movements," *Encyclopedia Human Brain*, vol. 2, pp. 399–414, 2002.

[45] S. L. Gorniak, V. M. Zatsiorsky, and M. L. Latash, "Manipulation of a fragile object," *Exp. Brain Res.*, vol. 202, no. 2, pp. 413–430, 2010.

[46] J. Park, T. Singh, V. M. Zatsiorsky, and M. L. Latash, "Optimality versus variability: Effect of fatigue in multi-finger redundant tasks," *Exp. Brain Res.*, vol. 216, no. 4, pp. 591–607, 2012.

[47] B. I. Prilutsky and V. M. Zatsiorsky, "Optimization-based models of muscle coordination," *Exerc. Sport Sci. Rev.*, vol. 30, no. 1, 2002, Art. no. 32.

[48] C. Fermüller, et al., "Prediction of manipulation actions," *Int. J. Comput. Vis.*, pp. 1–17, 2017, https://doi.org/10.1007/s11263-017-0992-z

[49] M. S. Lobo, L. Vandenberghe, S. P. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, pp. 193–228, 1998.

[50] S. P. Boyd and B. Wegbreit, "Fast computation of optimal contact forces," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1117–1132, Dec. 2007.

[51] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4–5, pp. 705–724, 2015.

[52] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, Jan. 2016.

[53] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[54] M. A. Brubaker, L. Sigal, and D. J. Fleet, "Estimating contact dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2389–2396.

[55] M. Mohammadi, T. L. Baldi, S. Scheggi, and D. Prattichizzo, "Fingertip force estimation via inertial and magnetic sensors in deformable object manipulation," in *Proc. IEEE Haptics Symp.*, 2016, pp. 284–289.

[56] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu, "Inferring forces and learning human utilities from videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3823–3833.

[57] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3D point clouds for indoor scenes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 244–252.

[58] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3050–3057.

[59] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 842–849.

[60] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[61] B. Çalli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015.

[62] C. Schedlinski and M. Link, "A survey of current inertia parameter identification methods," *Mech. Syst. Sign. Process.*, vol. 15, no. 1, pp. 189–211, 2001.

[63] K. S. Bhat, S. M. Seitz, J. Popović, and P. K. Khosla, "Computing the physical parameters of rigid-body motion from video," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 551–565.

[64] M. Ciocarlie, C. Lackner, and P. Allen, "Soft finger model with adaptive contact geometry for grasping and manipulation tasks," in *Proc. IEEE World Haptics Conf.*, 2007, pp. 219–224.

[65] D. Kraft, "Algorithm 733: Tomp-Fortran modules for optimal control calculations," *ACM Trans. Math. Softw.*, vol. 20, no. 3, pp. 262–281, 1994.

[66] J. L. Elman, "Finding structure in time," *Cognitive Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[68] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A Matlab-like environment for machine learning," in *Proc. BigLearn Neural Inf. Process. Syst. Workshop*, 2011.

[69] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[70] J. P. Scholz, F. Danion, M. L. Latash, and G. Schöner, "Understanding finger coordination through analysis of the structure of force variability," *Biol. Cybern.*, vol. 86, no. 1, pp. 29–39, 2002.

[71] N. Kyriazis and A. Argyros, "Scalable 3D tracking of multiple interacting objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3430–3437.

[72] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. San Mateo, CA, USA: Morgan Kaufmann, 2001.

[73] M. Mboup, C. Join, and M. Fliess, "Numerical differentiation with annihilators in noisy environment," *Numerical Algorithms*, vol. 50, no. 4, pp. 439–467, 2009.

[74] S. Stassi, V. Cauda, G. Canavese, and C. F. Pirri, "Flexible tactile sensing based on piezoresistive composites: A review," *Sensors*, vol. 14, no. 3, pp. 5296–5332, 2014.

[75] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A python package for convex optimization," 2013. [Online]. Available: abel.ee.ucla.edu/cvxopt

[76] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother, "6-DOF model based tracking via object coordinate regression," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 384–399.

[77] J. Issac, M. Wüthrich, C. Garcia Cifuentes, J. Bohg, S. Trimpe, and S. Schaal, "Depth-based object tracking using a robust gaussian filter," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 608–615.

[78] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Capturing and reproducing hand-object interactions through vision-based force sensing," in *Proc. IEEE ICCV Workshop Object Understanding Interaction*, 2015.

[79] Y. Zheng and K. Yamane, "Evaluation of grasp force efficiency considering hand configuration and using novel generalized penetration distance algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1580–1587.

[80] T.-H. Pham, S. Caron, and A. Kheddar, "Multi-contact interaction force sensing from whole-body motion capture," *IEEE Trans. Ind. Inform.*, 2017, doi: 10.1109/TII.2017.2760912.

Tu-Hoa Pham received the Dipl-Ing SupAéro degree from ISAE, the MSc degree in mathematics from the Université Paul Sabatier (France, 2013) and the PhD degree in robotics from Université de Montpellier, France, 2016, conducted between the CNRS-AIST Joint Robotics Laboratory, Japan, and CNRS-UM LIRMM, France. He is currently a postdoctoral researcher with IBM Research Tokyo. His research interests include robot vision and learning for monitoring of human activities and learning from demonstration.

Nikolaos Kyriazis received the BSc, MSc, and PhD diplomas from the University of Crete (UoC), in 2005, 2009, and 2014, respectively. He is currently a computer vision engineer with Oculus. His main research interests regard computational methods for observing and understanding the interaction of human and robotic systems with objects of their environment.

Antonis A. Argyros is a professor of computer science at the Computer Science Department, University of Crete and a researcher at the Institute of Computer Science, FORTH, in Heraklion, Crete, Greece. His research interests fall in the areas computer vision and pattern recognition, with emphasis on the analysis of humans in images and videos, human pose analysis, recognition of human activities and gestures, 3D computer vision, as well as image motion and tracking. He is also interested in applications of computer vision in the fields of robotics and smart environments. In these areas, he has published more than 150 papers in scientific journals and refereed conference proceedings.

Abderrahmane Kheddar (M'04, SM'12) is presently directeur de recherche with CNRS, Codirector of the CNRS-AIST Joint Robotic Laboratory, Japan, and leads the Interactive Digital Humans team at CNRS-UM LIRMM, France. His research interests include haptics, humanoids and thought-based control. He is a founding member of the IEEE/RAS chapter on haptics and of the *IEEE Transactions on Haptics*. He is a member of the steering committee of the IEEE Brain Initiative, editor of the *IEEE Transactions on Robotics* and within the editorial board of some other robotics journals. He is a senior member of the IEEE, titular member of the National Academy of Technology of France, and Knight in the National Order of the Merit.