Capturing Hands in Action using Discriminative Salient Points and Physics Simulation

Dimitrios Tzionas · Luca Ballan · Abhilash Srikantha · Pablo Aponte · Marc Pollefeys · Juergen Gall

Abstract Hand motion capture is a popular research field, recently gaining more attention due to the ubiquity of RGB-D sensors. However, even most recent approaches focus on the case of a single isolated hand. In this work, we focus on hands that interact with other hands or objects and present a framework that successfully captures motion in such interaction scenarios for both rigid and articulated objects. Our framework combines a generative model with discriminatively trained salient points to achieve a low tracking error and with collision detection and physics simulation to achieve physically plausible estimates even in case of occlusions and missing visual data. Since all components are unified in a single objective function which is almost everywhere differentiable, it can be optimized with standard optimization techniques. Our approach works for monocular RGB-D sequences as well as setups with multiple synchronized RGB cameras. For a qualitative and quantitative evaluation, we captured 29 sequences with a large variety of interactions and up to 150 degrees of freedom.

Keywords Hand motion capture · Hand-object interaction · Fingertip detection · Physics simulation

D. Tzionas · A. Srikantha · P. Aponte · J. Gall
Institute of Computer Science III, University of Bonn
Römerstraße 164, 53117 Bonn, Germany
E-mail: {tzionas,srikanth,aponte,gall}@iai.uni-bonn.de

D. Tzionas · A. Srikantha
Perceiving Systems Department
Max Planck institute for Intelligent Systems
Spemannstraße 41, 72076 Tübingen, Germany

L. Ballan · M. Pollefeys
Institute for Visual Computing, ETH Zurich
Universitätstraße 6, CH-8092 Zurich, Switzerland
E-mail: {luca.ballan,marc.pollefeys}@inf.ethz.ch

1 Introduction

Capturing 3d motion of human hands is an important research topic in computer vision since decades (Erol et al, 2007; Heap and Hogg, 1996) due to its importance for numerous applications including, but not limited to, computer graphics, animation, human computer interaction, rehabilitation and robotics. With recent technology advancements of consumer RGB-D sensors, the research interest in this topic has increased in the last few years (Tompson et al, 2014; Ye et al, 2013). Despite being a special instance of full human body tracking, it can not be easily solved by applying known techniques for human pose estimation like (Shotton et al, 2011) to human hands. While hands share some challenges with the full body like the high dimensionality of the underlying skeleton, they introduce additional difficulties. The body parts of the hands are very similar in shape and appearance, palm and forearm are difficult to model, and severe self-occlusions are a frequent phenomenon.

Due to these difficulties, the research from the first efforts in the field (Heap and Hogg, 1996) even until very recent approaches (Tompson et al, 2014) has mainly focused on a single isolated hand. While isolated hands are useful for a few applications like gesture control, humans use hands mainly for interacting with the environment and manipulating objects. In this work, we focus therefore on hands in action, i.e. hands that interact with other hands or objects. This problem has been addressed so far only by a few works (Hamer et al, 2009, 2010; Kyriazis and Argyros, 2013, 2014; Oikonomidis et al, 2011b, 2012, 2014). While Hamer et al (2009) considered objects only as occluders, Hamer et al (2010) derive a pose prior from the manipulated objects to support the hand tracking. This approach,

however, assumes that training data is available to learn the prior. A different approach to model interactions between objects and hands is based on a collision or physical model. Within a particle swarm optimization (PSO) framework, Oikonomidis et al (2011b, 2012) approximate the hand by spheres to detect and avoid collisions. In the same framework, Kyriazis and Argyros (2013, 2014) enrich the set of particles by using a physical simulation for hypothesizing the state of one or several rigid objects.

Instead of employing a sampling based optimization approach like PSO, we propose in this work a single objective function that combines data terms, which align the model with the observed data, with a collision and physical model. The advantage of our objective function is that it can be optimized with standard optimization techniques. In our experiments, we use local optimization and enrich the objective function with discriminatively learned salient points to avoid pose estimation errors due to local minima. Salient points, like finger tips, have been used in the earlier work of Rehg and Kanade (1994). Differently from their scenario, however, these salient points cannot be tracked continuously due to the huge amount of occlusions and the similarity in appearance of these features. Therefore we cannot rely on having a fixed association between the salient points and the respective fingers. To cope with this, we propose a novel approach that solves the salient point association jointly with the hand pose estimation problem.

Preliminary versions of this paper appeared in (Ballan et al, 2012; Tzionas et al, 2014). The present work unifies the pose estimation for multiple synchronized RGB cameras (Ballan et al, 2012) and a monocular RGB-D camera (Tzionas et al, 2014). In addition, the objective function is extended by a physical model that increases the realism and physical plausibility of the hand poses. In the experiments, we qualitatively and quantitatively evaluate our approach on 29 sequences and present for the first time successful tracking results of two hands strongly interacting with non-rigid objects.

2 Related Work

The study of hand motion tracking has its roots in the 90s (Rehg and Kanade, 1995, 1994). Although the problem can be simplified by means of data-gloves (Ekvall and Kragic, 2005), color-gloves (Wang and Popović, 2009), markers (Vaezi and Nekouie, 2011) or wearable sensors (Kim et al, 2012), the ideal solution pursued is the unintrusive, marker-less capture of hand motion.

In pursuit of this, one of the first hand tracking approaches (Rehg and Kanade, 1994) introduced the use

of local optimization in the field. Several filtering approaches have been presented (Bray et al, 2007; MacCormick and Isard, 2000; Stenger et al, 2001; Wu et al, 2001), while also belief-propagation proved to be suitable for articulated objects (Hamer et al, 2009, 2010; Sudderth et al, 2004). Oikonomidis et al (2011a) employ Particle Swarm Optimization (PSO) as a form of stochastic search, while later they present a novel evolutionary algorithm that capitalizes on quasi-random sampling (Oikonomidis et al, 2014). Kim et al (2012) and Wang and Popović (2009) use inverse-kinematics, while Heap and Hogg (1996) and Wu et al (2001) reduce the search space using linear subspaces. Athitsos and Sclaroff (2003) resort to probabilistic line matching, while Thayananthan et al (2003) combine Bayesian filtering with Chamfer matching. Recently, Schmidt et al (2014) extended the popular signed distance function (SDF) representation to articulated objects, while Qian et al (2014) combine a gradient based ICP approach with PSO, showing the complementary nature of the two approaches. Sridhar et al (2013) explore the use of a Sum of Gaussians (SoG) model for hand tracking on RGB images, which is later replaced by a Sum of Anisotropic Gaussians (Sridhar et al, 2014).

All these approaches have in common that they are generative models. They use an explicit model to generate pose hypotheses, which are evaluated against the observed data. The evaluation is based on an objective function which implicitly measures the likelihood by computing the discrepancy between the pose estimate (hypothesis) and the observed data in terms of an error metric. To keep the problem tractable, each iteration is initialized by the pose estimate of the previous step, relying thus heavily on temporal continuity and being prone to accumulative error. The objective function is evaluated in the high-dimensional, continuous parameter space. Recent approaches relax the assumption of a fixed predefined shape model, allowing for online non-rigid shape deformation (Taylor et al, 2014) that enables better data fitting and user-specific adaptation.

Discriminative methods learn a direct mapping from the observed image features to the discrete (Athitsos and Sclaroff, 2003; Romero et al, 2009, 2010) or continuous (de Campos and Murray, 2006; Rosales et al, 2001) target parameter space. Some approaches also segment the body parts first and estimate the pose in a second step (Keskin et al, 2012; Tompson et al, 2014). Most methods operate on a single frame, being thus immune to pose-drifting due to error accumulation. Generalization in terms of capturing illumination, articulation and view-point variation can be realized only through adequate representative training data. Acquisition and an-

Fig. 1 Qualitative results of our approach for the case of hand-hand interaction. Each pair shows the aligned RGB and depth input images after depth thresholding along with the pose estimate

notation of realistic training data is though a cumbersome and costly procedure. For this reason most approaches rely on synthetic rendered data (Keskin et al, 2012; Romero et al, 2010) that has inherent ground-truth. Special care is needed to avoid over-fitting to the training set, while the discrepancy between realistic and synthetic data is an important limiting factor. Recent approaches (Tang et al, 2013) tried to address the latter using transductive regression forests to transfer knowledge from fully labeled synthetic data to partially labeled realistic data. Finally, the accuracy of discriminative methods heavily depends on the invariance, repeatability and discriminative properties of the features employed and is lower in comparison to generative methods.

A discriminative method can effectively complement a generative method, either in terms of initialization or recovery, driving the optimization framework away from local minima in the search space and aiding convergence to the global minimum. Sridhar et al (2013) combine in a real time system a Sum of Gaussians (SoG) generative model with a discriminatively trained fingertip detector in depth images using a linear SVM classifier. Alternatively, the model can also be combined with a part classifier based on random forests (Sridhar et al, 2015). Recently, Sharp et al (2015) combined a PSO optimizer with a robust, two-stage regression reinitializer that predicts a distribution over hand poses from a single RGB-D frame.

Generative and discriminative methods have used various low level image cues for hand tracking that are often combined, namely silhouettes, edges, shading, color, optical flow (de La Gorce et al, 2011; Lu et al, 2003), while depth (Bray et al, 2007; Delamarre and Faugeras, 2001; Hamer et al, 2009) has recently gained popularity with the ubiquity of RGB-D sensors (Oikonomidis et al, 2011a, 2012; Qian et al, 2014; Schmidt et al, 2014; Sridhar et al, 2013; Tompson et al, 2014). In this work, we combine in a single framework a generative model with discriminative salient points detected by a Hough forest (Gall et al, 2011b), i.e. a finger nail detector on color images and finger tip detector on depth images, respectively. As low level cues,

we use edges and optical flow for the RGB sequences and depth for the RGB-D sequences.

3 Pose Estimation

Our approach for capturing the motion of hands and manipulated objects can be applied to RGB-D and multi-view RGB sequences. In both cases hands and objects are modeled in the same way as described in Section 3.1. The main difference between RGB-D and RGB sequences is the used data term, which depends on depth or edges and optical flow, respectively. We therefore introduce first the objective function for a monocular RGB-D sequence in Section 3.2 and describe the differences for RGB sequences in Section 3.3.

3.1 Hand and Object Models

We resort to the popular linear blend skinning (LBS) model (Lewis et al, 2000), consisting of a triangular mesh with an underlying kinematic skeleton, as depicted in Figure 2a-c, and a set of skinning weights. In our experiments, a triangular mesh of a pair of hands was obtained by a 3D scanning solution, while meshes for several objects (ball, cube, pipe, rope) were created manually with a 3D graphics software. Some objects are shown in Figure 3. A skeletal structure defining the kinematic chain was manually defined and fitted into the meshes. The skinning weight $\kappa_{\mathbf{v},j}$ defines the influence of bone j on 3D vertex $\mathbf{v}$, where $\sum_j \kappa_{\mathbf{v},j} = 1$. Figure 4 visualizes the mesh using the largest skinning weight for each vertex as bone association. The deformation of each mesh is driven by its underlying skeleton with pose parameter vector θ through the skinning weights and is expressed by the LBS operator:

$$\mathbf{v}(\theta) = \sum_j \kappa_{\mathbf{v},j} T_j(\theta) T_j(0)^{-1} \mathbf{v}(0) \tag{1}$$

where $T_j(0)$ and $\mathbf{v}(0)$ are the bone transformations and vertex positions at the known rigging pose. The skinning weights are computed using (Baran and Popović, 2007).
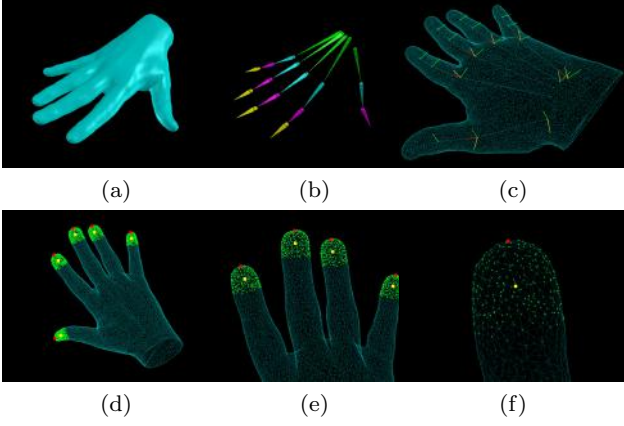
Fig. 2 Hand model used for tracking. (a) Mesh (b) Kinematic Skeleton (c) Degrees of Freedom (DoF) (d-f) Mesh fingertips (green) used for the salient point detector. The vertices of the fingertips are found based on the manually annotated red vertices. The centroid of the fingertips, as defined in Section 3.2.5, is depicted with yellow color
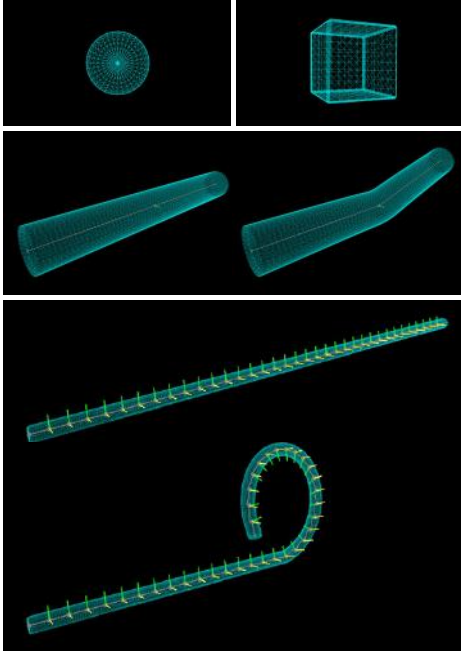

Fig. 3 Object models used for tracking and their DoF: (top-left) a rigid ball with 6 DoF; (top-right) a rigid cube with 6 DoF; (middle) a pipe with 1 revolute joint, i.e. 7 DoF; (bottom) a rope with 70 revolute joints, i.e. 76 DoF

The global rigid motion is represented by a 6 DoF twist $\vartheta\xi = \vartheta(u_1, u_2, u_3, \omega_1, \omega_2, \omega_3)$ with $\|\omega\| = 1$ (Bregler et al, 2004; Murray et al, 1994; Pons-Moll and Rosenhahn, 2011). The twist action $\vartheta\hat{\xi} \in se(3)$ has the
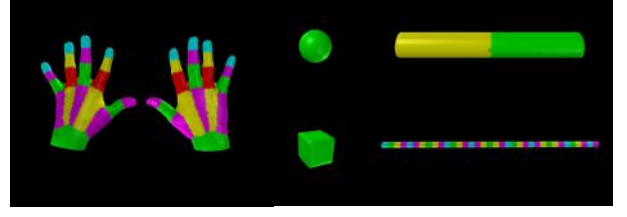

Fig. 4 Segmentation of the meshes based on the skinning weights. The ball and the cube are rigid objects while the pipe and rope are modeled as articulated objects. Each hand has 20 skinning bones, the pipe has 2, while the rope has 36

form of a 4×4 matrix

$$\vartheta\hat{\xi} = \vartheta \begin{pmatrix} \hat{\omega} & u \\ 0_{1\times3} & 0 \end{pmatrix} = \vartheta \begin{pmatrix} 0 & -\omega_3 & \omega_2 & u_1 \\ \omega_3 & 0 & -\omega_1 & u_2 \\ -\omega_2 & \omega_1 & 0 & u_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

and the exponential map operator $\exp(\vartheta\hat{\xi})$ defines the group action:

$$T(\vartheta\hat{\xi}) = \begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix} = \exp(\vartheta\hat{\xi}) \in SE(3). \quad (3)$$

While $\theta = \vartheta\xi$ for a rigid object, articulated objects have additional parameters. We model the joints by revolute joints. A joint with one DoF is modeled by a single revolute joint, i.e. the transformation of the corresponding bone j is given by $\exp(\vartheta_{p(j)}\xi_{p(j)})\exp(\vartheta_j\xi_j)$ where $p(j)$ denotes the parent bone. If a bone does not has a parent bone, it is the global rigid transformation. The transformation of an object with one revolute joint is thus described by $\theta = (\vartheta\xi, \vartheta_1)$. Joints with two or three DoF are modeled by a combination of K_j revolute joints, i.e. $\prod_{k=1}^{K_j} \exp(\vartheta_{j,k}\xi_{j,k})$. For simplicity, we denote the relative transformation of a bone j by $\hat{T}_j(\theta) = \prod_{k=1}^{K_j} \exp(\vartheta_{j,k}\xi_{j,k})$. The global transformation of a bone j is then recursively defined by

$$T_j(\theta) = T_{p(j)}(\theta)\hat{T}_j(\theta). \quad (4)$$

In our experiments, a single hand consists of 31 revolute joints, i.e. 37 DoF, as shown in Figure 2c. The rigid objects have 6 DoF. The deformations of the non-rigid shapes shown in Figure 3 are approximated by a skeleton. The pipe has 1 revolute joint, i.e. 7 DoF, while the rope has 70 revolute joints, i.e. 76 DoF. Thus, for sequences with two interacting hands we have to estimate all 74 DoF and together with the rope 150 DoF.

3.2 Objective Function

Our objective function for pose estimation consists of seven terms:

$$
\begin{aligned}
E(\theta, D) = \ & E_{model \to data}(\theta, D) + E_{data \to model}(\theta, D) + \\
& \gamma_c E_{collision}(\theta) + E_{salient}(\theta, D) + \\
& \gamma_{ph} E_{physics}(\theta) + \gamma_a E_{anatomy}(\theta) + \\
& \gamma_r E_{regularization}(\theta)
\end{aligned}
\tag{5}
$$

where θ are the pose parameters of the template meshes and D is the current preprocessed depth image. The preproccesing is explained in Section 3.2.1. The first two terms minimize the alignment error of the transformed mesh and the depth data. The alignment error is measured by $E_{model \to data}$, which measures how well the model fits the observed depth data, and $E_{data \to model}$, which measures how well the depth data is explained by the model. $E_{salient}$ measures the consistency of the generative model with detected salient points in the image. The main purpose of the term in our framework is to recover from tracking errors of the generative model. $E_{collision}$ penalizes intersections of fingers and $E_{physics}$ enhances the realism of grasping poses during interaction with objects. Both of the terms $E_{collision}$ and $E_{physics}$ ensure physically plausible poses and are complementary. The term $E_{anatomy}$ enforces amatomically inspired joint limits, while the last term is a simple regularization term that prefers the solution of the previous frame if there are insufficient oberservations to determine the pose.

In the following, we give details for the terms of the objective function (5) as well as the optimization of it.

3.2.1 Preprocessing:

For pose estimation, we first remove irrelevant parts of the RGB-D image by thresholding the depth values, in order to avoid unnecessary processing like normal computation for points far away. Segmentation of the hand from the arm is not necessary and is therefore not performed. Subsequently we apply skin color segmentation on the RGB image (Jones and Rehg, 2002). As a result we get masked RGB-D images, denoted as D in (5). The skin color segmentation separates hands and non-skin colored objects, facilitating hand and object tracking accordingly.

3.2.2 Fitting the model to the data - LO_{m2d}:

The first term in Equation (5) aims at fitting the mesh parameterized by pose parameters θ to the preprocessed data D. To this end, the depth values are converted into a 3D point cloud based on the calibration data of the sensor. The point cloud is then smoothed by a bilateral filter (Paris and Durand, 2009) and normals are computed (Holzer et al, 2012). For each visible vertex of the model $\mathbf{v}_i(\theta)$, with normal $\mathbf{n}_i(\theta)$, we search for the closest point X_i in the point cloud. This gives a 3D-3D correspondence for each vertex. We discard the correspondence if the angle between the normals of the vertex and the closest point is larger than $45°$ or the distance between the points is larger than 10 mm. We can then write the term $E_{model \to data}$ as a least squared error of *point-to-point* distances:

$$
E_{model \to data}(\theta, D) = \sum_i \| \mathbf{v}_i(\theta) - X_i \|^2
\tag{6}
$$

An alternative to the *point-to-point* distance is the *point-to-plane* distance, which is commonly used for 3D reconstruction (Chen and Medioni, 1991; Rusinkiewicz and Levoy, 2001; Rusinkiewicz et al, 2002). In this case:

$$
E_{model \to data}(\theta, D) = \sum_i \| \mathbf{n}_i(\theta)^T (\mathbf{v}_i(\theta) - X_i) \|^2 \ .
\tag{7}
$$

The two distance metrics are evaluated in Section 4.1.1.

3.2.3 Fitting the data to the model - LO_{d2m}:

Only fitting the model to the data is not sufficient as we will show in our experiments. In particular, poses with self-occlusions can have a very low error since the measure only evaluates how well the visible part of the model fits the point cloud. The second term $E_{data \to model}(\theta, D)$ matches the data to the model to make sure that the solution is not degenerate and explains the data as well as possible. However, matching the data to the model is more expensive since after each iteration the pose changes, which would require to update the data structure for matching, e.g. distance fields or kd-trees, after each iteration. We therefore reduce the matching to depth discontinuities (Gall et al, 2011a). To this end, we extract depth discontinuities from the depth map and the projected depth profile of the model using an edge detector (Canny, 1986). Correspondences are again established by searching for the closest points, but now in the depth image using a 2D distance transform (Felzenszwalb and Huttenlocher, 2004). Similar to $E_{model \to data}(\theta, D)$, we discard correspondences with a large distance. The depth values at the depth discontinuities in D, however, are less reliable not only due to the depth ambiguities between foreground and background, but also due to the noise of consumer sensors. The depth of the point in D is therefore computed as
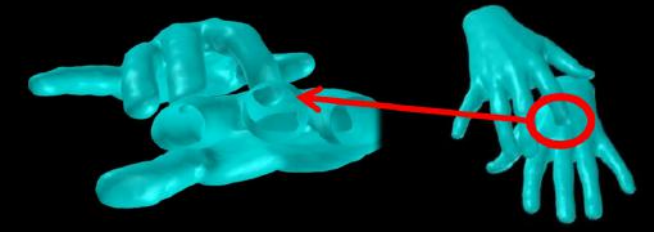
Fig. 5 "Walking" sequence. Without the collision term unrealistic mesh intersections are observed during interactions

average in a local 3×3 pixels neighborhood and the outlier distance threshold is increased to 30 mm. The approximation is sufficient for discarding outliers, but insufficient for minimization. For each matched point in D we therefore compute the projection ray uniquely expressed as a Plücker line (Pons-Moll and Rosenhahn, 2011; Rosenhahn et al, 2007; Stolfi, 1991) with direction $\mathbf{d}_i$ and moment $\mathbf{m}_i$ and minimize the least square error between the projection ray and the vertex $\mathbf{v}_i(\theta)$ for each correspondence:

$$E_{data \rightarrow model}(\theta, D) = \sum_i \|\mathbf{v}_i(\theta) \times \mathbf{d}_i - \mathbf{m}_i\|^2 \qquad (8)$$

We compared the matching based on depth discontinuities with a direct matching of the point cloud to the model using a kd-tree. The direct matching increases the runtime by 40% or more without reducing the error.

3.2.4 Collision detection - $\mathcal{C}$

Collision detection is based on the observation that two objects cannot share the same space and is of high importance in case of self-penetration, inter-finger penetration or general intensive interaction, as in the case depicted in Figure 5.

Collisions between meshes are detected by efficiently finding the set of colliding trianges $\mathcal{C}$ using bounding volume hierarchies (BVH) (Teschner et al, 2004). In order to penalize collisions and penetrations, we avoid using a signed 3D distance field for the whole mesh due to its high computational complexity and the fact that it has to be recomputed at every iteration of the optimization framework. Instead, we resort to a more efficient approach with local 3D distance fields defined by the set of colliding triangles $\mathcal{C}$ that have the form of a cone as depicted in Figure 6. In case of multiple collisions the defined conic distance fields are summed up as shown in the same figure. Having found a collision between two triangles f_t and f_s, the amount of penetration can be computed by the position inside the conic distance fields. The value of the distance field represents

the intention of the repulsion that is needed to penalize the intrusion.

Let us consider the case where the vertices of f_t are the *intruders* and the triangle f_s is the *receiver* of the penetration. The opposite case is then similar. The cone for computing the 3D distance field $\Psi_{f_s} : \mathbb{R}^3 \rightarrow \mathbb{R}_+$ is defined by the circumcenter of the triangle f_s. Letting $\mathbf{n}_{f_s} \in \mathbb{R}^3$ denote the normal of the triangle, $\mathbf{o}_{f_s} \in \mathbb{R}^3$ the circumcenter, and $r_{f_s} \in \mathbb{R}_{>0}$ the radius of the circumcircle, we have

$$\Psi_{f_s}(\mathbf{v}_t) = \begin{cases} |(1 - \Phi(\mathbf{v}_t)) \Upsilon(\mathbf{n}_{f_s} \cdot (\mathbf{v}_t - \mathbf{o}_{f_s}))|^2 & \Phi(\mathbf{v}_t) < 1 \\ 0 & \Phi(\mathbf{v}_t) \geq 1 \end{cases}$$

$$(9)$$

$$\Phi(\mathbf{v}_t) = \frac{\|(\mathbf{v}_t - \mathbf{o}_{f_s}) - (\mathbf{n}_{f_s} \cdot (\mathbf{v}_t - \mathbf{o}_{f_s}))\mathbf{n}_{f_s}\|}{-\frac{r_{f_s}}{\sigma}(\mathbf{n}_{f_s} \cdot (\mathbf{v}_t - \mathbf{o}_{f_s})) + r_{f_s}} \qquad (10)$$

$$\Upsilon(x) = \begin{cases} -x + 1 - \sigma & x \leq -\sigma \\ -\frac{1-2\sigma}{4\sigma^2}x^2 - \frac{1}{2\sigma}x + \frac{1}{4}(3 - 2\sigma) & x \in (-\sigma, +\sigma) \\ 0 & x \geq +\sigma. \end{cases}$$

$$(11)$$

The term Φ projects the vertex $\mathbf{v}$ onto the axis of the right circular cone defined by the triangle normal $\mathbf{n}$ going through the circumcenter $\mathbf{o}$ and measures the distance to it as illustrated in Figure 6. The distance is scaled by the radius of the cone at this point. If $\Phi(\mathbf{v}) < 1$ the vertex is inside the cone and if $\Phi(\mathbf{v}) = 0$ the vertex is on the axis. The term Υ measures how far the projected point is from the circumcenter and defines the intensity of the repulsion. If $\Upsilon < 0$, the projected point is behind the triangle. Within the range $(-\sigma, +\sigma)$, the penalizer term is quadratic with values between zero and one. If the penetration is larger than $|\sigma|$ the penalizer term becomes linear. The parameter σ also defines the field of view of the cone and is fixed to 0.5.

For each vertex penetrating a triangle, a repulsion term in the form of a 3D-3D correspondence that pushes the vertex back is computed. The direction of the repulsion is given by the inverse normal direction of the vertex and its intensity by Ψ. Using *point-to-point* distances, the repulsion correspondences are computed for the set of colliding triangles $\mathcal{C}$:

$$E_{collision}(\theta) = \sum_{(f_s(\theta), f_t(\theta)) \in \mathcal{C}} \left\{ \sum_{\mathbf{v}_s \in f_s} \| -\Psi_{f_t}(\mathbf{v}_s)\mathbf{n}_s\|^2 + \right.$$

$$\left. \sum_{\mathbf{v}_t \in f_t} \| -\Psi_{f_s}(\mathbf{v}_t)\mathbf{n}_t\|^2 \right\}$$
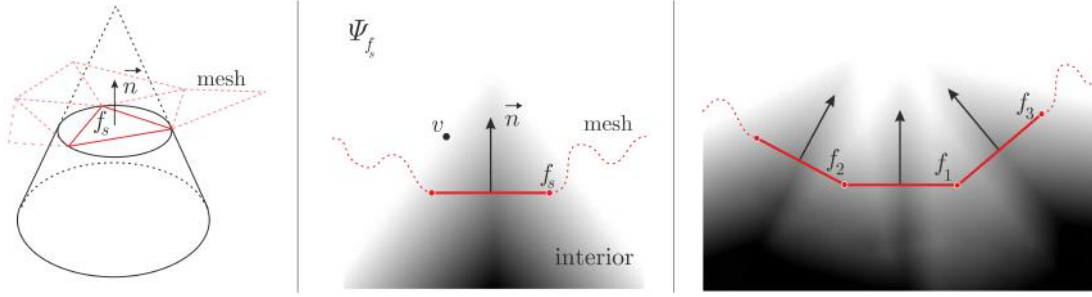
Fig. 6 (Left) Domain of the distance field Ψ_{f_s} generated by the face f_s. (Middle) Longitudinal section of the distance field Ψ_{f_s}: darker areas correspond to higher penalties. (Right) Distance fields add up in case of multiple collisions

(12)

Though not explicitly denoted, f_s and f_t depend on θ and therefore also Ψ, $\mathbf{v}$ and $\mathbf{n}$. For *point-to-plane* distances, the equation gets simplified since $\mathbf{n}^T \mathbf{n} = 1$:

$$E_{collision}(\theta) = \sum_{(f_s(\theta), f_t(\theta)) \in \mathcal{C}} \left\{ \sum_{\mathbf{v}_s \in f_s} \| - \Psi_{f_t}(\mathbf{v}_s) \|^2 + \sum_{\mathbf{v}_t \in f_t} \| - \Psi_{f_s}(\mathbf{v}_t) \|^2 \right\}$$

(13)

This term takes part in the objective function (5) regulated by weight γ_c. An evaluation of different γ_c values is presented in Section 4.1.3.

3.2.5 Salient point detection - $\mathcal{S}$

Our approach is so far based on a generative model, which provides accurate solutions in principle, but recovers only slowly from ambiguities and tracking errors. However, this can be compensated by integrating a discriminatively trained salient point detector into a generative model.

To this end, we train a fingertip detector on raw depth data. We manually annotate[1] the fingertips of 56 sequences consisting of approximately 2000 frames, with 32 of the sequences forming the training and 24 forming the testing set. We use a Hough forest (Gall et al, 2011b) with 10 trees, each trained with 100000 positive and 100000 negative patches. The negative patches are uniformly sampled from the background. The trees have a maximal depth of 25 and a minimum leaf size of 20 samples. Each patch is sized 16×16 and consists of 11 feature channels: 2 channels obtained by a 5×5 min- and max-filtered depth channel and 9 gradient features obtained by 9 HOG bins using a 5×5 cell and soft binning. As for the pool of split functions at a test node, we

---

[1] All annotated sequences are available at `http://files.is.tue.mpg.de/dtzionas/hand-object-capture.html`

randomly generate a set of 20000 binary tests. Testing is performed at multiple scales and non-maximum suppression is used to retain the most confident detections that do not overlap by more than 50%.

Since we resort to salient points only for additional robustness, it is usually sufficient to have only sparse fingertip detections. We therefore collect detections with a high confidence, choosing a threshold of $c_{thr} = 3.0$ for our experiments. The association between the T fingertips ϕ_t of the model depicted in Figure 2 (d-f) and the S detections δ_s is solved by integer programming (Belongie et al, 2002):

$$\begin{aligned} \operatorname*{arg\,min}_{e_{st}, \alpha_s, \beta_t} \quad & \sum_{s,t} e_{st} w_{st} + \lambda \sum_s \alpha_s w_s + \lambda \sum_t \beta_t \\ \text{subject to} \quad & \sum_s e_{st} + \beta_t = 1 \qquad \forall t \in \{1, ..., T\} \\ & \sum_t e_{st} + \alpha_s = 1 \qquad \forall s \in \{1, ..., S\} \\ & e_{st}, \alpha_s, \beta_t \in \{0, 1\} \end{aligned}$$

(14)

As illustrated in Table 1, $e_{st} = 1$ defines an assignment of a detection δ_s to a fingertip ϕ_t. The assignment cost is defined by w_{st}. If $\alpha_s = 1$, the detection δ_s is declared as a false positive with cost λw_s and if $\beta_t = 1$, the fingertip ϕ_t is not assigned to any detection with cost λ.

The weights w_{st} are given by the 3D distance between the detection δ_s and the finger of the model ϕ_t. For each finger ϕ_t, a set of vertices are marked in the model. The distance is then computed between the 3D centroid of the visible vertices of ϕ_t (Figure 2d-f) and the centroid of the detected region δ_s. The latter is computed based on the 3D point cloud δ'_s corresponding to the detection bounding box. For the weights w_s, we investigate two approaches. The first approach uses $w_s = 1$. The second approach takes the confidences c_s of the detections into account by setting $w_s = \frac{c_s}{c_{thr}}$. The weighting parameter λ is evaluated in Section 4.1.2.

Table 1 The graph contains T mesh fingertips ϕ_t and S fingertip detections δ_s. The cost of assigning a detection δ_s to a fingertip ϕ_t is given by w_{st} as shown in table (a). The cost of declaring a detection as false positive is λw_s, where w_s is the detection confidence. The cost of not assigning any detection to fingertip ϕ_t is given by λ. The binary solution of table (b) is constrained to sum up to one for each row and column

(a)		Fingertips ϕ_t				V
		ϕ_1	ϕ_2	$\ldots$	ϕ_T	α
Detections δ_s	δ_1	w_{11}	w_{12}	$\ldots$	w_{1T}	λw_1
	δ_2	w_{21}	w_{22}	$\ldots$	w_{2T}	λw_2
	δ_3	w_{31}	w_{32}	$\ldots$	w_{3T}	λw_3
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	δ_S	w_{S1}	w_{S2}	$\ldots$	w_{ST}	λw_S
V	β	λ	λ	$\ldots$	λ	∞

(b)		Fingertips ϕ_t				V
		ϕ_1	ϕ_2	$\ldots$	ϕ_T	α
Detections δ_s	δ_1	e_{11}	e_{12}	$\ldots$	e_{1T}	α_1
	δ_2	e_{21}	e_{22}	$\ldots$	e_{2T}	α_2
	δ_3	e_{31}	e_{32}	$\ldots$	e_{3T}	α_3
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	δ_S	e_{S1}	e_{S2}	$\ldots$	e_{ST}	α_S
V	β	β_1	β_2	$\ldots$	β_T	0

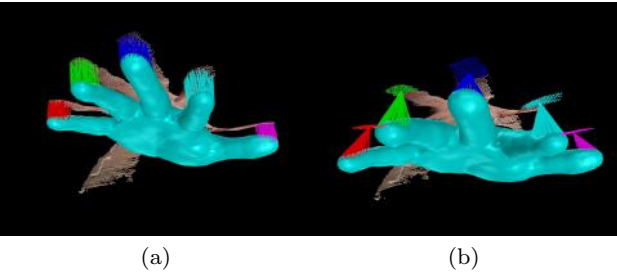(a)   (b)

Fig. 7 Correspondences between the fingertips ϕ_t of the model and (a) the closest points of the associated detections δ'_s (b) the centroids of the associated detections δ'_s

(a)   (b)

(c)   (d)

Fig. 8 Physical plausibility during hand-object interaction. (a) Input RGB-D image. (b-c) Obtained results without the physics component. (d) Obtained results with the physics component, ensuring a more realistic pose during interaction

If a detection δ_s has been associated to a fingertip ϕ_t, we have to define correspondences between the set of visible vertices of ϕ_t and the detection point cloud δ'_s. If the fingertip ϕ_t is already very close to δ'_s, i.e. $w_{st} < 10mm$, we do not compute any correspondences since the localization accuracy of the detector is not higher. In this case just the close proximity of the fingertip ϕ_t to the data suffices for a good alignment. Otherwise, we compute the closest points between the vertices $\mathbf{v}_i$ and the points X_i of the detection δ'_s as illustrated in Figure 7a:

$$E_{salient}(\theta, D) = \sum_{s,t} e_{st} \left(\sum_{(X_i,\mathbf{v}_i) \in \delta'_s \times \phi_t} \|\mathbf{v}_i(\theta) - X_i\|^2 \right)$$
(15)

As in (7), a *point-to-plane* distance metric can replace the *point-to-point* metric. When less than 50% of the vertices of ϕ_t project inside the detection bounding box, we even avoid the additional step of computing correspondences between the vertices and the detection point cloud. Instead we associate all vertices with the centroid of the detection point cloud as shown in Figure 7b.
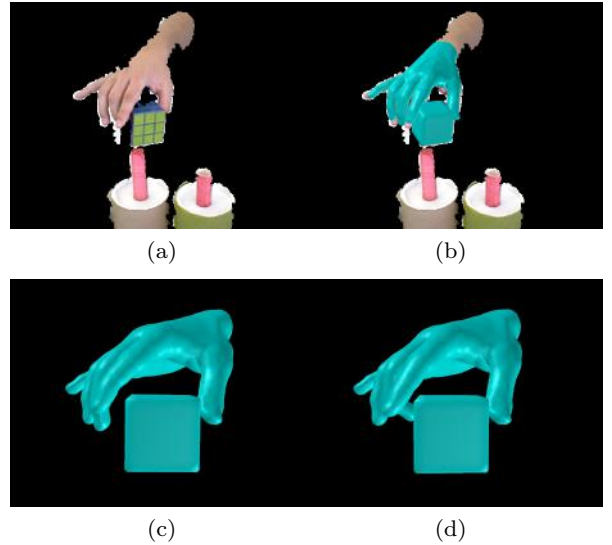
3.2.6 Physics Simulation - $\mathcal{P}$

A phenomenon that frequently occurs in the context of hand-object interaction are physically unrealistic poses due to occlusions or missing data. Such an example is illustrated in Figure 8, where a cube is grasped and moved by two fingers. Since one of the fingers that is in contact with the cube is occluded, the estimated pose is physical unrealistic. Due to gravity, the cube would fall down.

In order to compensate for this during hand-object interaction scenarios, we resort to physics simulation (Coumans, 2013) for additional realism and physical plausibility. To this end, we model the static scene as well and based on gravity and the parameters friction, restitution, and mass for each object we can run a physics simulation. To speed up the simulation, we represent each body or object part defined by the skinning

weights as shown in Figure 4 as convex hulls. This is visualized in Figure 9.

Given current pose estimates of the hands and the manipulated object, we first evaluate if the current solution is physically plausible. To this end, we run the simulation for 35 iterations with a time-step of 0.1 seconds. If the centroid of the object moves by less than 3mm we consider the solution as stable. Otherwise, we have to search for the hand pose which results in a more stable estimate. Since it is intractable to evaluate all possible hand poses, we search only for configurations which require a minor change of the hand pose. This is a reasonable assumption for our tracking scenario. To this end, we first compute the distances between all parts of the fingers, as depicted in Figure 10, and the object (Aggarwal et al, 1987; Gärtner and Schönherr, 2000). Each finger part with distance less than 10mm is then considered as candidate for being in contact with the object and each combination of at least two and maximum four candidate parts is taken into account.

The contribution of each combination to the stability of the object is examined through the physics simulation after rigidly moving the corresponding finger parts towards the closest surface point of the object. Figure 9 illustrates the case for a combination of two finger parts. The simulation is repeated for all combinations and we select the combination with the lowest movement of the object, i.e. the smallest displacement of its centroid from the initial position. Based on the selected combination, we define an energy that forces the corresponding finger parts to get in contact with the object by minimizing the closest distance between the parts i and the object:

$$E_{physics}(\theta) = \sum_i \|\mathbf{v}_i(\theta) - X_i\|^2 \qquad (16)$$

The vertices $\mathbf{v}_i$ and X_i correspond to the closest point of a finger part and the object, respectively. As in (7), a *point-to-plane* distance metric can replace the *point-to-point* metric.

3.2.7 Anatomical limits

Anatomically inspired joint-angle limits (Albrecht et al, 2003) are enforced as soft constraints by the term:

$$E_{anatomy}(\theta) = \sum_k \left(\exp\left(p(l_k - \theta_k)\right) + \exp\left(p(\theta_k - u_k)\right)\right) \qquad (17)$$

where $p = 10$. The index k goes over all revolute joints and $[u_k, l_k]$ is the allowed range for each of them. The term is illustrated for a single revolute joint in Figure 11. We use $\gamma_a = 0.0015\, C_{all}$, where C_{all} is the total number of correspondences.
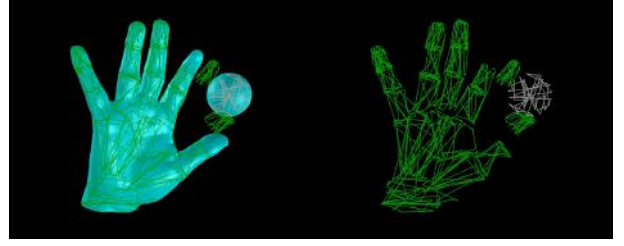

Fig. 9 Low resolution representation of the hands and objects for the physics simulation. In order to predict the finger parts (green) that give the physically most stable results if they were in contact with the object (white), all combinations of finger parts close to the object are evaluated. The image shows how two finger parts are moved to the object for examining the contribution to the stability of the object. The stability is measured by a physics simulation where all green parts are static

Fig. 10 Finger parts that form all possible supporting combinations in the physics simulation component. Parts with red color do not take part in this process

(a)   (b)
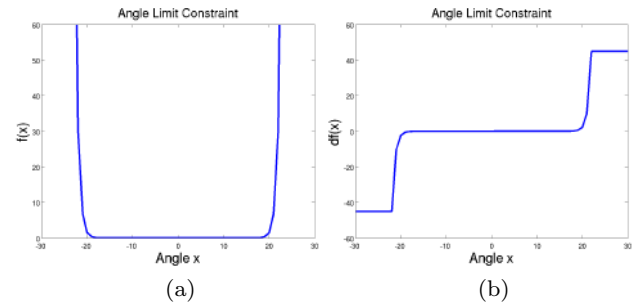
Fig. 11 Angle limits are independently defined for each revolute joint. The plot visualizes the function (a), and its truncated derivative (b), that penalizes the deviation from an allowed range of ± 20.0 degree

3.2.8 Regularization

In case of occlusions or due to missing depth data, the objective function (5) based solely on the previous terms can be ill-posed. We therefore add a term that penalizes deviations from the previous estimated joint angles $\tilde{\theta}$:

$$E_{regularization}(\theta) = \sum_k (\theta_k - \tilde{\theta}_k)^2. \qquad (18)$$

We use $\gamma_r = 0.02\, C_{all}$.

Algorithm 1: Pose estimation for RGB-D data
with *point-to-point* distances

$\tilde{\theta}$ = pose estimate of the previous frame
$i = 0$, $\theta_0 = \tilde{\theta}$
Repeat until convergence or max i_{thr} iterations
  - Render meshes at pose θ
  - Find corresp. LO_{m2d} Section 3.2.2 - Eq. (6)
  - Find corresp. LO_{d2m} Section 3.2.3 - Eq. (8)
  - Find corresp. $\mathcal{C}$ Section 3.2.4 - Eq. (12)
  - Find corresp. $\mathcal{S}$ Section 3.2.5 - Eq. (15)
  - Find corresp. $\mathcal{P}$ Section 3.2.6 - Eq. (16)
  $\theta_{i+1} = \arg\min_\theta E(\theta, D)$
  $i = i + 1$

3.2.9 Optimization

For pose estimation, we alternate between computing
the correspondences LO_{m2d} (Section 3.2.2), LO_{d2m} (Section 3.2.3), $\mathcal{C}$ (Section 3.2.4), $\mathcal{S}$ (Section 3.2.5), and
$\mathcal{P}$ (Section 3.2.6) according to the current pose estimate and optimizing the objective function (5) based
on them as summarized in Algorithm 1. This process is
repeated until convergence or until a maximum number of iterations i_{thr} is reached. It should be noted that
the objective function $E(\theta, D)$ is only differentiable for
a given set of correspondences. We optimize $E(\theta, D)$,
which is a non-linear least squares problem, with the
Gauss-Newton method as in (Brox et al, 2010).

3.3 Multicamera RGB

The previously described approach can also be applied
to multiple synchronized RGB videos. To this end, the
objective function (5) needs to be changed only slightly
due to the differences of depth and RGB data. While
the error is directly minimized in 3D for RGB-D data,
we minimize the error for RGB images in 2D since all
our observations are 2D. Instead of using a 3D *point-to-point* (6) or *point-to-plane* (7) measure, the error is
therefore given by

$$\sum_c \sum_i \| \Pi_c(\mathbf{v}_i(\theta)) - x_{i,c} \|^2 \tag{19}$$

where $\Pi_c : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$ are the known projection functions, mapping 3D points into the image plane of each
static camera c, and $(\mathbf{v}_i, x_{i,c})$ is a correspondence between a 3D vertex and a 2D point. Furthermore, the
salient point detector, introduced in Section 3.2.5, is not
applied to the depth data but to all camera views. Since
multiple high resolution views allow to detect more distinctive image features, we do not detect finger tips but
finger nails in this case.

The only major change is required for the data terms
$E_{model \to data}(\theta, D)$ and $E_{data \to model}(\theta, D)$ in (5). The
term $E_{data \to model}(\theta, D)$ is replaced by an edge term
that matches edge pixels in all camera views to the
edges of the projected model in the current pose θ. As
in the RGB-D case, the orientation of the edges is taken
into account for matching and mismatches are removed
by thresholding. Working with 2D distances though has
the disadvantage of not being able to apply intuitive
3D distance thresholds, as presented in Section 3.2.3.
In order to have an alternative rejection mechanism of
noisy correspondences, we compute for each bone the
standard deviation of the 2D error that is suggested
by all of its correspondences. Subsequently, correspondences that suggest an error bigger than twice this standard deviation are rejected as outliers. The second term
$E_{model \to data}(\theta, D)$ is replaced by a term based on optical flow as in (Ballan and Cortelazzo, 2008). The term
introduces temporal consistency and harness the higher
resolution and frame rates of the RGB data in comparison to the RGB-D data.

4 Experimental Evaluation

Benchmarking in the context of 3D hand tracking remains an open problem (Erol et al, 2007) despite recent contributions (Qian et al, 2014; Sridhar et al, 2013;
Tang et al, 2013, 2014; Tompson et al, 2014; Tzionas
and Gall, 2013; Tzionas et al, 2014). The vast majority
of them focuses on the problem of single hand tracking, especially in the context of real-time human computer interaction, neglecting challenges occurring during the interaction between two hands or between hands
and objects. For this reason we captured 29 sequences
in the context of hand-hand and hand-object interaction. The sequences were captured either with a single
RGB-D camera or with 8 synchronized RGB cameras.
While 20 sequences have been used in the preliminary
works (Ballan et al, 2012; Tzionas et al, 2014), which
include interactions with rigid objects, the 9 newly captured sequences also include interactions with non-rigid
objects.

We first evaluate our approach on RGB-D sequences
with hand-hand interactions in Section 4.1. Sequences
with hand-object interactions are used in Section 4.2
for evaluation and finally our approach is evaluated on
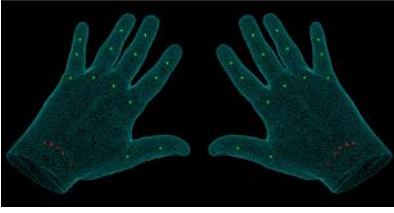sequences captured with several RGB cameras in Section 4.4.

Fig. 12 Hand joints used for quantitative evaluation. Only the green joints of our hand model are used for measuring the pose estimation error

4.1 Monocular RGB-D - Hand-Hand Interactions

Related RGB-D methods (Oikonomidis et al, 2011a) usually report quantitative results only on synthetic sequences, which inherently include ground-truth, while for realistic conditions they resort to qualitative results.

Although qualitative results are informative, quantitative evaluation based on ground-truth is of high importance. We therefore manually annotated 14 sequences, 11 of which are used to evaluate the components of our pipeline and 3 for comparison with the state-of-the-art method (Oikonomidis et al, 2011a). These sequences contain motions of a single hand and two interacting hands with 37 and 74 DoF, respectively. They vary from 100 to 270 frames and contain several actions, like *"Walking"*, *"Crossing"*, *"Crossing and Twisting"*, *"Tips Touching"*, *"Dancing"*, *"Tips Blending"*, *"Hugging"*, *"Grasping"*, *"Flying"*, as well as performing the *"Rock"* and *"Bunny"* gestures. As indicator for the accuracy of the annotations, we measured the standard deviation of 4 annotators, which is 1.46 pixels. All sequences were captured in $640x480$ resolution at 30 fps with a Primesense Carmine 1.09 camera.

The error metric for our experiments is the 2D distance (pixels) between the projection of the 3D joints and the corresponding 2D annotations. The joints taken into account in the metric are depicted in Figure 12. Unless explicitly stated, we report the average over all frames of all relevant sequences.

Our system is based on an objective function consisting of five terms, described in Section 3.2. Two of them minimize the error between the posed mesh and the depth data by fitting the *model to the data* and the *data to the model*. A *salient point* detector further constrains the pose using fingertip detections in the depth image, while a *collision detection* method contributes to realistic pose estimates that are physically plausible. The function is complemented by the *physics simulation* component, that contributes towards more realistic interaction of hands with objects. However, this component is only relevant for hand-object interactions and thus it will be studied in detail in Section 4.2. In the

Table 2 Evaluation of *point-to-point* (*p2p*) and *point-to-plane* (*p2plane*) distance metrics, along with iterations number of the optimization framework, using a 2D distance error metric (px). The highlighted setting is used for all other experiments

Iterations	5	10	15	20	30
p2p	7.33	5.25	5.05	4.98	4.91
p2plane	5.33	5.12	5.08	5.07	5.05

Table 3 Evaluation of the weighting parameter λ in (14), using a 2D distance error metric (px). Weight $\lambda = 0$ corresponds to the objective function without salient points, noted as "$LO+C$" in Table 6. Both versions of w_s described in Section 3.2.5 are evaluated. The highlighted setting is used for all other experiments

λ	*0*	0.3	0.6	0.9	1.2	1.5	1.8
$w_s = 1$	5.17	5.17	5.15	5.14	5.12	5.12	5.23
$w_s = \frac{c_s}{c_{thr}}$		5.14	5.12	5.12	5.12	5.22	5.61

following, we evaluate each component and the parameters of the objective function (5).

4.1.1 Distance Metrics

Table 2 presents an evaluation of the two distance metrics presented in Section 3.2.2, namely *point-to-point* (6) and *point-to-plane* (7), along with the number of iterations of the minimization framework. The *point-to-plane* metric leads to an adequate pose estimation error with only 10 iterations, providing a significant speed gain compared to *point-to-point*. If the number of iterations does not matter, the *point-to-point* metric is preferable since it results in a lower error and does not suffer from wrongly estimated normals.

For the first frame, we perform 50 iterations in order to ensure an accurate refinement of the manually initialized pose. For the chosen setup, we measure the runtime for the sequence *"Bunny"* that contains one hand and for the sequence *"Crossing and Twisting"* that contains two hands. For the first sequence, the runtime is 2.82 seconds per frame, of which 0.12 seconds are attributed to the salient point component $\mathcal{S}$ and 0.65 to the collision component $\mathcal{C}$. For the second sequence, the runtime is 4.96 seconds per frame, of which 0.05 seconds are attributed to the component $\mathcal{S}$ and 0.36 to the component $\mathcal{C}$.

4.1.2 Salient Point Detection - $\mathcal{S}$

The salient point detection component depends on the parameters w_s and λ, as described in Section 3.2.5. Table 3 summarizes our evaluation of the parameter λ spanning a range of possible values for both cases
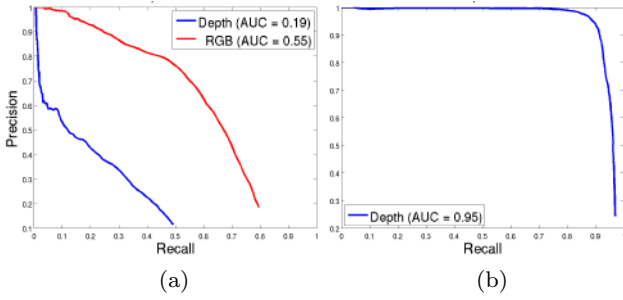
(a)   (b)

Fig. 13 Precision-recall plot for (a) our RGB-D dataset and (b) the Dexter dataset. We show the performance of a fingertip detector trained only on depth (blue) and only rgb (red) images. The area under the curve (AUC) for our dataset (a) is 0.19 and 0.55 respectively. The AUC for the Dexter dataset (b) is 0.95

Table 4 Evaluation of collision weights γ_c, using a 2D distance error metric (px). Weight 0 corresponds to the objective function without collision term, noted as "$LO + \mathcal{S}$" in Table 6. Sequences are grouped in 3 categories: "*severe*" for intense, "*some*" for light and "*no apparent*" for imperceptible collision. "$\geq some$" is the union of "*severe*" and "*some*". The highlighted value is the default value we use for all other experiments

γ_c	0	1	2	3	5	7.5	10	12.5
All	5.34	5.44	5.57	5.16	5.12	5.12	5.12	5.14
Severe	5.90	6.07	6.27	5.62	5.56	5.57	5.55	5.61
$\geq Some$	5.44	5.57	5.72	5.23	5.18	5.19	5.18	5.22
Some	3.99	3.98	3.98	3.98	3.98	3.99	3.99	3.98

$w_s = 1$ and $w_s = \frac{c_s}{c_{thr}}$. The differences between the two versions of w_s is minor although the optimal range of λ varies for the two versions. The latter is expected since $\frac{c_s}{c_{thr}} \geq 1$ and smaller values of λ compensate for the mean difference to $w_s = 1$ in (14). If $\lambda = 0$ all detections are classified as false positives and the salient points are not used in the objective function (5).

    To evaluate the performance of the detector, we follow the PASCAL-VOC protocol (Everingham et al, 2010). Figure 13a shows the precision-recall plot for our RGB-D dataset including all hand-hand and hand-object sequences. The plot shows that the detector does not perform well on this dataset and suffers from the noisy raw depth data. This also explains why the salient term improves the pose estimation only slightly. We therefore trained and evaluated the detector also on the RGB data. In this case, the detection accuracy is much higher. We also evaluated the detector on the Dexter dataset (Sridhar et al, 2013). On this dataset, the detector is very accurate. Our experiments on Dexter in Section 4.1.6 and a multi-camera RGB dataset in Section 4.4 will show that the salient points reduce the error more if the detector performs better.

Table 5 Comparison of the proposed collision term based on 3D distance fields with correspondences between vertices of colliding triangles

	Corresponding vertices	Distance fields
All	6.66	5.12
Severe	7.96	5.55
$\geq$ Some	7.04	5.18
Some	4.12	3.99

4.1.3 Collision Detection - $\mathcal{C}$

The impact of the collision detection component is regulated in the objective function (5) by the weight γ_c. For the evaluation, we split the sequences in three sets depending on the amount of observed collision: *severe*, *some*, and *no apparent* collision. The set with *severe* collisions comprises *"Walking"*, *"Crossing"*, *"Crossing and Twisting"*, *"Dancing"*, *"Hugging"*, *some* collisions are present in *"Tips Touching"*, *"Rock"*, *"Bunny"*, and no collisions are apparent in *"Grasping"*, *"Tips Blending"*, *"Flying"*. Table 4 summarizes our evaluation experiments for the values of γ_c. The results show that over all sequences, the collision term reduces the error and that a weight $\gamma_c \geq 3$ gives similar results. For small weights $0 < \gamma_c < 3$, the error is even slightly increased compared to $\gamma_c = 0$. In this case, the impact is too small to avoid collisions and the term only adds noise to the pose estimation. As expected, the impact of the collision term is only observed for the sequences with *severe* collision.

    The proposed collision term is based on a fast approximation of the distance field of an object. It is continuous and less sensitive to a change of the mesh resolution than a repulsion term based on 3D-3D correspondences between vertices of colliding triangles. To show this, we replaced the collision term by correspondences that move vertices of colliding triangles towards the counterpart. The results in Table 5 show that such a simple repulsion term performs poorly.

4.1.4 Component Evaluation

Table 6 presents the evaluation of each component and the combination thereof. Simplified versions of the pipeline, fitting either just the *model to the data* (LO_{m2d}) or the *data to the model* (LO_{d2m}) can lead to a collapse of the pose estimation, due to unconstrained optimization. Our experiments quantitatively show the notable contribution of both the collision detection and the salient point detector. The best overall system performance is achieved with all four components of the objective function (5). The fifth term $E_{physics}$ is only

Table 6 Evaluation of the components of our pipeline. "*LO*" stands for local optimization and includes fitting both *data-to-model* (*d2m*) and *model-to-data* (*m2d*), unless otherwise specified. Collision detection is noted as "*C*", while salient point detector is noted as "*S*". The number of sequences where the optimization framework collapses is noted in the last row, while the mean error is reported only for the rest

Components	LO_{m2d}	LO_{d2m}	LO	$LO + C$	$LO + S$	$LO + CS$
Mean Error (px)	27.17	–	5.53	5.17	5.34	5.12
Improvement (%)			–	6.46	3.44	7.44
Failed Sequences	1/11	11/11	0/11	0/11	0/11	0/11

Table 7 Pose estimation error for each sequence

	Walking	Crossing	Crossing Twisting	Tips Touching	Dancing	Tips Blending	Hugging	Grasping	Flying	Rock	Bunny
Mean Error (px)	5.99	4.53	4.76	3.65	6.49	4.87	5.22	4.37	5.11	4.44	4.50
Standard Deviation (px)	3.65	2.99	3.51	2.21	3.70	2.97	3.42	2.06	2.77	2.63	2.61
Max Error (px)	24.19	18.03	22.80	13.60	20.25	18.36	20.03	11.05	15.03	14.76	10.63

Fig. 14 Qualitative comparison with (Oikonomidis et al, 2011a). Each image pair corresponds to the pose estimate of the FORTH tracker (up) and our tracker (down)

relevant for hand-object interactions and will be evaluated in Section 4.2. Table 7 shows the error for each sequence. Figure 18, which is at the end of the article, depicts qualitative results for 8 out of the 11 sequences. It shows that the hand motion is accurately captured even in cases of close interaction and severe occlusions. The data and videos are available.[2]

4.1.5 Comparison to State-of-the-Art

Recently, Oikonomidis et al (2011a,b, 2012) used particle swarm optimization (PSO) for a real-time hand tracker. For comparison we use the software released for

---

[2]All annotated sequences are available at `http://files.is.tue.mpg.de/dtzionas/hand-object-capture.html`.

tracking a single hand (Oikonomidis et al, 2011a), with the parameter setups used also in the other works. Each setup is evaluated three times in order to compensate for the manual initialization and the inherent randomness of PSO. Qualitative results depict the best result of all three runs, while quantitative results report the average error. Table 8 shows that our system outperforms (Oikonomidis et al, 2011a) in terms of tracking accuracy. Figure 14 shows a visual comparison. However, it should be noted that the GPU implementation of (Oikonomidis et al, 2011a) runs in real time using 25 generations and 64 particles, in contrast to our single-threaded CPU implementation.

4.1.6 Dexter dataset

We further evaluate our approach on the recently introduced Dexter dataset (Sridhar et al, 2013). As suggested in (Sridhar et al, 2013), we use the first part of the sequences for evaluation and the second part for training. More specifically, the evaluation set contains the frames $018-158$ of the sequence "adbadd", $061-185$ of "fingercount", $020-173$ of "fingerwave", $025-224$ of "flexex1", $024-148$ of "pinch", $024-123$ of "random", and $016-166$ of "tigergrasp". We use only the depth of the Time-of-Flight camera.

The performance of our tracker is summarized in Tables 9 and 10. Since the dataset does not provide a hand model, we simply scaled our hand model in (x, y, z) direction by $(0.95, 0.95, 1)$. Since the annotations in the dataset do not correspond to anatomical landmarks but are close to the finger tips, we compare the annotations with the endpoints of our skeleton. Table 9 shows the error of our tracker for each of the sequences, reporting the mean, the maximum, and the standard deviation of the error over all the tested frames. Despite of the

Table 8 Comparison with (Oikonomidis et al, 2011a). We evaluate the FORTH tracker with 4 parameter settings, 3 of which were used in the referenced literature of the last column

		Mean (px)	St.Dev (px)	Max (px)	Generations	Particles	Reference
FORTH	*set 1*	8.58	5.74	61.81	25	64	Oikonomidis et al (2011a)
	set 2	8.32	5.42	57.97	40	64	Oikonomidis et al (2011b)
	set 3	8.09	5.00	38.90	40	128	
	set 4	8.16	5.18	39.85	45	64	Oikonomidis et al (2012)
Proposed		3.76	2.22	19.92			

Table 9 Pose estimation error of our tracker for each sequence of the Dexter dataset.

LO + $\mathcal{SC}$	Mean Error	St. Deviation	Max Error	
Adbadd	17.34	15.35	69.73	[mm]
Fingercount	11.94	7.18	47.77	
Fingerwave	10.88	5.47	49.62	
Flexex1	11.87	12.86	91.70	
Pinch	24.19	28.34	131.97	
Random	96.93	122.34	559.37	
Tigergrasp	11.77	5.36	30.18	
Adbadd	7.79	8.38	42.54	[px]
Fingercount	6.03	5.39	38.28	
Fingerwave	4.45	2.80	15.26	
Flexex1	5.24	8.37	61.40	
Pinch	12.56	16.48	73.16	
Random	59.93	77.77	307.00	
Tigergrasp	6.84	4.22	21.21	

Table 10 Evaluation of the components of the objective function (5) on the Dexter dataset. "*LO*" stands for local optimization and includes fitting both *data-to-model* and *model-to-data*. Collision detection is noted as "$\mathcal{C}$" and salient point detector as "$\mathcal{S}$". The "random" sequence is excluded because our approach fails due to very fast motion

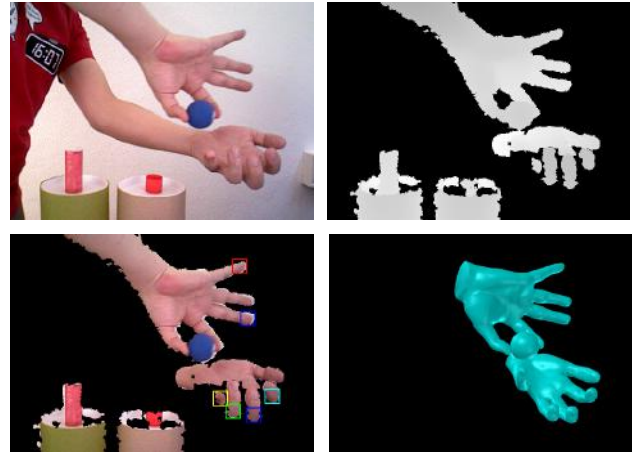| Components	Mean Error	St. Dev.	
LO + $\mathcal{SC}$	14.26	14.91	[mm]
LO + $\mathcal{S}$	15.51	16.67	
LO + $\mathcal{C}$	16.97	16.60	
LO	17.86	18.80	
LO + $\mathcal{SC}$	6.90	8.88	[px]
LO + $\mathcal{S}$	7.64	9.87	
LO + $\mathcal{C}$	8.98	10.29	
LO	9.33	10.73	

Fig. 15 Failure case due to missing data and detection errors. The images show RGB image (top-left), input depth image (top-right), fingertip detections (bottom-left), and estimated pose (bottom-right). The detector operates on the raw depth image, while the RGB image is used just for visualization

differences of our hand model and the data, the average error is for most sequences only around 1*cm*. Our approach, however, fails for the sequence "random" due to the very fast motion in the sequence.

Table 10 presents the evaluation of each component of our pipeline and the combination of them. On this dataset, both the collision term as well as the salient point detector reduce the error. Compared to Table 6, the error is more reduced. In particular, the salient point detector reduces the error more since the detector performs well on this dataset as shown in Figure 13b. Compared to "*LO*", the average error of "*LO + SC*" is by more than 3.5mm lower. The average error reported by Sridhar et al (2013) on the slow part of the Dexter dataset is 13.1 mm.

4.2 Monocular RGBD - Hand-Object Interactions

For the evaluation of the complete energy function (5) for hand-object interactions, we captured 7 new sequences[2] of hands interacting with several objects, either rigid (*ball, cube)* or articulated (*pipe, rope*). The DoF of the objects varies a lot. The rigid objects have 6 DoF, the *pipe* 7 DoF, and the *rope* 76 DoF. The sequences vary from 180 to 400 frames and contain several actions, like: "*Moving a Ball*" with one (43 DoF) or two hands (80 DoF), "*Moving a Cube*" with one hand (43 DoF), "*Bending a Pipe*" with two hands (81 DoF), and "*Bending a Rope*" with two hands (150 DoF). In addition, the sequences "*Moving a Ball*" with one hand and "*Moving a Cube*" were captured twice, one with occlu-

Table 11 Evaluation of the friction value of both the hands and the object. We report the error over all the frames of all seven sequences with hand-object interactions using a 2d error metric (px). Value 3.0 is the same as the friction value of the static scene. The highlighted value is the default value we use for all other experiments

Friction	0.6	0.9	1.2	1.5	3.0	
Mean	6.19	6.18	6.19	6.17	6.17	[px]
St. Dev.	3.82	3.81	3.81	3.81	3.81	

Table 12 Evaluation of collision weights γ_{ph} for "$LO + \mathcal{SCP}$", using a 2D distance error metric (px). Weight 0 corresponds to the objective function without physics term, noted as "$LO + \mathcal{SC}$" in Table 13. Sequences are grouped in 3 categories: "*severe*" for intense, "*some*" for light and "*no apparent*" for imperceptible occlusion of manipulating fingers. "$\geq$ *some*" is the union of "*severe*" and "*some*". The highlighted value is the default value we use for all other experiments

γ_{ph}	0	1	2	3	5	7.5	10	12.5
All	6.21	6.20	6.21	6.19	6.19	6.18	6.19	6.17
Severe	5.68	5.66	5.65	5.63	5.63	5.63	5.62	5.61
$\geq$ *Some*	6.02	6.00	6.00	5.98	5.98	5.97	5.96	5.94

sion of a manipulating finger and one without. Manual ground-truth annotation was performed by a single subject.

For the salient point ($\mathcal{S}$) and the collision detection component ($\mathcal{C}$), we use the parameter setup presented in Section 4.1. The influence of the physics simulation component ($\mathcal{P}$) and its parameters are evaluated in the following section. The error metric used is the 2D distance (pixel units) between the projection of the 3D joints and the 2D annotations as in Section 4.1 and visualized in Figure 12. Unless otherwise stated, we report the average over all frames of all seven sequences.

4.2.1 Physics Simulation - $\mathcal{P}$

For the physics simulation, we model the entire scene, which includes the hands as well as manipulated and static objects, with a low resolution representation as described in Section 3.2.6 and visualized in Figure 9. Each component of the scene is characterized by three properties: friction, restitution, and mass. Since in each simulation step we consider each component except of the manipulated object as static, only the mass of the object is relevant, which we set to 1 kg. We set the restitution of the static scene and hands to 0 and of the object to 0.5. For the static scene, we use a friction value of 3. The friction for both the hand and the object are assumed to be equal. Since the main purpose of the physics simulation is to evaluate if the current pose estimates are physical stable, the exact values for friction, restitution, and mass are not crucial. To demonstrate this, we evaluate the impact of the friction value for hands and manipulated objects. For this experiment, we set the weight γ_{ph} equal to 10.0, being the same as the weight γ_c of the complementary collision detection component. The results presented in Table 11 show that the actual value of friction has no significant impact on the pose estimation error as long as it is in a reasonable range.

The impact of the physics simulation component $E_{physics}$ in the objective function (5) is regulated by the weight γ_{ph}. The term penalizes implausible manipulation or grasping poses. For the evaluation, we split

the sequences in three sets depending on the amount of occlusions of the manipulating fingers: "*severe*" for intense ("Moving a Cube" with one hand and occlusion), "*some*" for light ("Moving a Ball" with one hand and occlusion, "Moving a Cube" with one hand) and "*no apparent*" for imperceptible occlusions (*"Moving a Ball"* with one and two hands, *"Bending a Pipe"*, *"Bending a Rope"*). Table 12 summarizes the pose estimation error for various values of γ_{ph} for the three subsets. Although the pose estimation error is only slightly reduced by $E_{physics}$, the results are physically more plausible. This is shown in Figure 19 at the end of the article, which provides a qualitative comparison between the setups "LO + $\mathcal{SC}$" and "LO + $\mathcal{SCP}$". The images show the notable contribution of component $\mathcal{P}$ towards more realistic, physically plausible poses, especially in cases of missing or ambiguous visual data, as in sequences with an occluded manipulating finger. To quantify this, we run the simulation for 35 iterations with a time-step of 0.1 seconds after the pose estimation and measured the displacement of the centroid of the object for each frame. While the average displacement is 9.26mm for the setup "LO + $\mathcal{SC}$", the displacement is reduced to 9.05mm by the setup "LO + $\mathcal{SCP}$". The tracking runtime for the aforementioned sequences for the setup "LO + $\mathcal{SC}$" ranges from 4 to 8 seconds per frame. The addition of $\mathcal{P}$ in the setup "LO + $\mathcal{SCP}$" increases the runtime for most sequences for about 1 second. However, this increase might reach up to more than 1 minute depending on the complexity of the object and tightness of interaction, as in the case of *"Bending a Pipe"* with two hands (150 DoF), with the main bottleneck being the computation of the closest finger vertices to the manipulated object. Figure 20 depicts qualitative results for the full setup "LO + $\mathcal{SCP}$" of the objective function (5) for all seven sequences. The results show successful tracking of interacting hands with both rigid and articulated objects, whose articulation is described from 1 to as many as 71 DoF.

Table 13 Evaluation of the components of the objective function (5). "*LO*" stands for local optimization and includes fitting both *data-to-model* and *model-to-data*. Collision detection is noted as "$\mathcal{C}$", salient point detector as "$\mathcal{S}$" and physics simulation as "$\mathcal{P}$". We report the error for fixed 10 iterations and for the stopping criterion $\epsilon < 0.2mm$

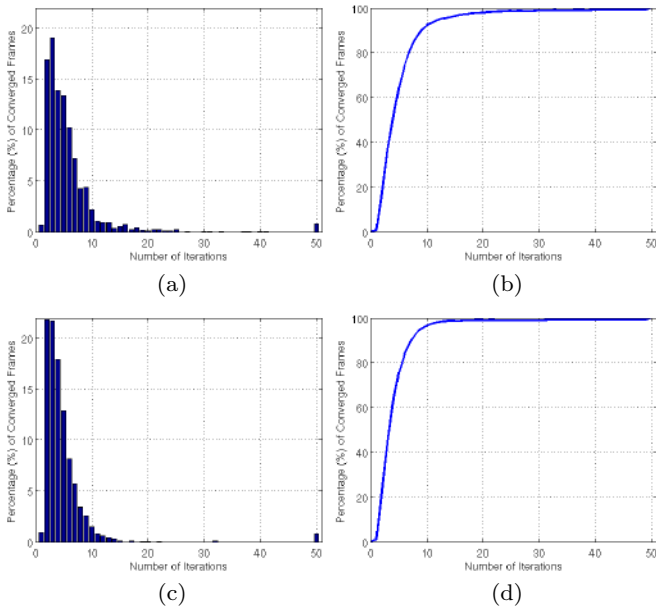| Components	Fixed 10 Iterations		Stopping Thresh. 0.2 mm	
	Mean	St. Dev.	Mean	St. Dev.
LO + $\mathcal{SCP}$	6.19	3.81	6.25	3.86
LO + $\mathcal{SC}$	6.21	3.82	6.31	3.89
LO + $\mathcal{S}$	6.05	3.76	6.09	3.77
LO + $\mathcal{CP}$	6.19	3.83	6.31	3.90
LO + $\mathcal{C}$	6.24	3.84	6.38	3.94
LO	6.07	3.77	6.15	3.83
	px		px	

Fig. 16 Number of iterations that are required to converge for LO + $\mathcal{SCP}$ (top) and LO (bottom). (a,c) Distribution of frames where the pose estimation converged after a given number of iterations. (b,d) Cumulative distribution

4.2.2 Component Evaluation

Table 13 presents the evaluation of each component and their combinations for the seven sequences with hand-object interaction. Since the physical simulations $\mathcal{P}$ assumes that there are no severe intersections, it is meaningful only as a complement to the collision component $\mathcal{C}$. One can observe that the differences between the components are relatively small since the hand poses in the hand-object sequences are in general simpler than the poses in the sequences with tight hand-hand interactions as considered in Section 4.1. The collision term

$\mathcal{C}$ slightly increases the error, but without the term the hand poses are often physically implausible and intersect with the object. When comparing LO + $\mathcal{SC}$ and LO + $\mathcal{SCP}$, we see that the error is slightly reduced by the physics simulation component $\mathcal{P}$. The pose estimation errors for each sequence using LO + $\mathcal{SCP}$ are summarized in Table 14.

Instead of using a fixed number of iterations per frame, a stopping criterion can be used. We use the average change of the joint positions after each iteration. As threshold, we use $0.2mm$ and a maximum of 50 iterations. Table 13 shows that for the stopping criterion the impact of the terms is slightly more prominent, but it also shows that the error is slightly higher for all approaches. To analyze this more in detail, we report the distribution of required iterations until the stopping criterion is reached in Figure 16. Although LO + $\mathcal{SCP}$ requires a few more iterations until convergence compared to LO, it converges in 10 or less iterations in 92% of the frames, which supports our previous results. There are, however, very few frames where the approach has not converged after 50 iterations. In most of these cases, the local optimum of the energy is far away from the true pose and the error is increased with more iterations. These outliers are also the reason for the slight increase of the error in Table 13. For all combinations from LO to LO + $\mathcal{SCP}$ we observed this behavior, which shows that the energy can be further improved.

4.3 Limitations

As shown in Sections 4.1 and 4.2, our approach captures accurately the motion of hands tightly interacting either with each other or with a rigid or articulated object. However, for very fast motion like the "random" sequence of the Dexter dataset our approach fails. Furthermore, we assume that a hand model is given or can be acquired by an approach like (Taylor et al, 2014). Figure 15 also visualizes an inaccurate hand pose of the lower hand due to missing depth data and two detections, which are not at the finger tips but located at other bones.

4.4 Multicamera RGB

We finally evaluated the approach for sequences captured using a setup of 8 synchronized cameras recording FullHD footage at 50 fps. To this end, we recorded 9 sequences that span a variety of hand-hand and hand-object interactions, namely: *"Praying"*, *"Fingertips Touching"*, *"Fingertips Crossing"*, *"Fingers Crossing and Twisting"*, *"Fingers Folding*, *"Fingers Walk-*

Table 14 Pose estimation error for each sequence

	Moving Ball 1 hand	Moving Ball 2 hands	Bending Pipe	Bending Rope	Moving Ball 1 hand, occlusion	Moving Cube 1 hand	Moving Cube 1 hand, occlusion	
Mean Error	6.10	7.15	6.09	5.65	8.03	4.68	5.55	[px]
Standard Deviation	3.90	4.82	3.07	3.04	5.47	2.61	3.28	

ing" on the back of the hand, *"Holding and Passing a Ball"*, *"Paper Folding"* and *"Rope Folding"*. The length of the sequences varies from 180 to 1500 frames.

Figure 21 shows one frame from each of the tested sequences and the obtained results overlaid on the original frames from two different cameras. Visual inspection reveals that the proposed algorithm works also quite well for multiple RGB cameras even in challenging scenarios of very closely interacting hands with multiple occlusions. The data and videos are available.[3]

4.4.1 Component Evaluation

As for the RGB-D sequences, we also evaluate the components of our approach. To this end, we synthesized two sequences: first, fingers crossing and folding, and second, holding and passing a ball, both similar to the ones captured in the real scenario. Videos were generated using a commercial rendering software. The pose estimation accuracy was then evaluated both in terms of error in the joints position, and in terms of error in the bones orientation.
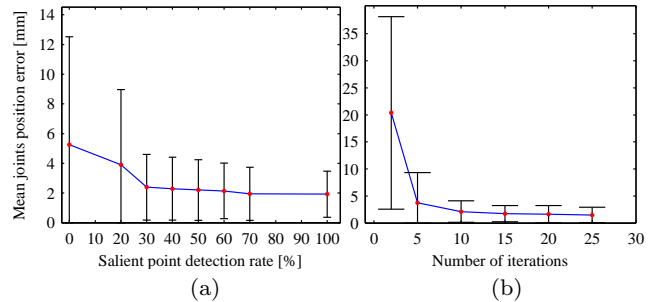
Table 15 shows a quantitative evaluation of the algorithm performance with respect to the used visual features. It can be noted that each feature contributes to the accuracy of the algorithm and that the salient points $\mathcal{S}$ clearly boost its performance. The benefit of the salient points is larger than for the RGB-D sequences since the localization of the finger tips from several high-resolution RGB cameras is more accurate than from a monocular depth camera with lower resolution. This is also indicated by the precision-recall curves in Figure 13a.

We also compared with (Oikonomidis et al, 2011a) on the synthetic data where we used an own implementation since the publicly available source code requires a single RGB-D sequence. We also added the salient points term and used two settings, namely 64 and 128 particles over 40 generations. The results in Table 15 show that our approach estimates the pose with a lower error and confirm the results for the RGB-D sequences reported in Table 8.

Table 15 Quantitative evaluation of the algorithm performance with respect to the used visual features: edges $\mathcal{E}$, collisions $\mathcal{C}$, optical flow $\mathcal{O}$, and salient points $\mathcal{S}$. LO stands for our local optimization approach, while HOPE64 and HOPE128 stand for our implementation of (Oikonomidis et al, 2011a) with 64 and 128 particles respectively, evaluated over 40 generations.

Used features	Mean	St.Dev.	Max	
LO + $\mathcal{E}$	3.11	4.52	49.86	
LO + $\mathcal{EC}$	2.50	2.89	52.94	
LO + $\mathcal{ECO}$	2.38	2.25	16.84	[mm]
LO + $\mathcal{ECOS}$	1.49	1.44	13.27	
HOPE64 + $\mathcal{ECOS}$	4.86	3.69	31.05	
HOPE128 + $\mathcal{ECOS}$	4.67	3.28	41.11	

Used features	Mean	St.Dev.	Max	
LO + $\mathcal{E}$	2.36	6.84	94.58	
LO + $\mathcal{EC}$	1.98	4.57	91.89	
LO + $\mathcal{ECO}$	1.84	3.81	60.09	[deg]
LO + $\mathcal{ECOS}$	1.88	3.90	44.51	
HOPE64 + $\mathcal{ECOS}$	4.35	7.11	58.61	
HOPE128 + $\mathcal{ECOS}$	4.73	7.46	78.65	

Fig. 17 Quantitative evaluation of the algorithm performance on noisy data, with respect to the salient point detection rate (a), and the number of iterations (b). Black bars indicate the standard deviation of the obtained error.

In order to make the synthetic experiments as realistic as possible, we simulated noise in all of the visual features. More precisely, edge detection errors were introduced by adding structural noise to the images, i.e. by adding and subtracting at random positions in each image 100 circles of radius varying between 10 and 30 pixels. The optical flow features corresponding to those circles were also not considered. Errors in the salient point detector were simulated by randomly deleting detections as well as by randomly adding outliers in a

Table 16 Results obtained on the manually marked data for the multicamera RGB sequences. The table reports the distance in mm between the manually tracked 3D points and the corresponding vertices on the hand model. The figure shows the positions of the tracked points on the hand.

Points	median	mean	std	max
point 1	06.98	07.98	3.54	20.53
point 2	11.14	12.28	5.22	23.48
point 3	10.91	10.72	4.13	24.68

radius of 200 pixels around the actual features. Gaussian noise of 5 pixels was further introduced on the coordinates of the resulting salient points. Figure 17(a) shows the influence of the salient point detector on the accuracy of the pose estimation in case of noisy data. This experiment was run with a salient point false positive rate of 10%, and with varying detection rates. It is visible that the error quickly drops very close to its minimum even with a detection rate of only 30%.

Figure 17(b) shows the convergence rate for different numbers of iterations. It can be noted that the algorithm accuracy becomes quite reasonable after just $10 - 15$ iterations, which is the same as for the RGB-D sequences.

We also annotated one of the captured sequences for evaluation. Since annotating joints in multiple RGB cameras is more time consuming than annotating joints in a single RGB-D camera, we manually labeled only three points on the hands in all camera views of the sequence *"Holding and Passing a Ball"*. Since we obtain 3D points by triangulation, we therefore use the 3D distance between these points and the corresponding vertices in the hand model as error metric. Table 16 shows the tracking accuracy obtained in this experiment. Overall, the median of the tracking error is at maximum 1cm.

5 Conclusion

In this paper we have presented a framework that captures articulated motion of hands and manipulated objects from monocular RGB-D videos as well as multiple synchronized RGB videos. Contrary to works that focus on gestures and single hands, we focus on the more difficult case of intense hand-hand and hand-object interactions. To address the difficulties, we have proposed an approach that combines in a single objective function a generative model with discriminatively trained salient points, collision detection and physics simulation. Although the collision and physics term reduce the pose estimation only slightly, they increase the realism of the captured motion, especially under occlusions and missing visual data. We performed qualitative and quantitative evaluations on 8 sequences captured with multiple RGB cameras and on 21 sequences captured with a single RGB-D camera. Comparisons with an approach based on particle swarm optimization (Oikonomidis et al, 2011a) for both camera systems revealed the our model achieves a higher accuracy for hand pose estimation. For the first time, we present successful tracking results of hands interacting with highly articulated objects.

6 Acknowledgements

References

Aggarwal A, Klawe MM, Moran S, Shor P, Wilber R (1987) Geometric applications of a matrix-searching algorithm. Algorithmica 2(1-4):195–208

Albrecht I, Haber J, Seidel HP (2003) Construction and animation of anatomically based human hand models. In: SCA, pp 98–109

Athitsos V, Sclaroff S (2003) Estimating 3d hand pose from a cluttered image. In: CVPR, pp 432–439

Ballan L, Cortelazzo GM (2008) Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: 3DPVT

Ballan L, Taneja A, Gall J, Van Gool L, Pollefeys M (2012) Motion capture of hands in action using discriminative salient points. In: ECCV, pp 640–653

Baran I, Popović J (2007) Automatic rigging and animation of 3d characters. TOG 26(3)

Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. PAMI 24(4):509–522

Bray M, Koller-Meier E, Van Gool L (2007) Smart particle filtering for high-dimensional tracking. CVIU 106(1):116–129

Bregler C, Malik J, Pullen K (2004) Twist based acquisition and tracking of animal and human kinematics. IJCV 56(3):179–194

Brox T, Rosenhahn B, Gall J, Cremers D (2010) Combined region- and motion-based 3d tracking of rigid and articulated objects. PAMI 32(3):402–415

de Campos T, Murray D (2006) Regression-based hand pose estimation from multiple cameras. In: CVPR

Canny J (1986) A computational approach to edge detection. PAMI 8(6):679–698

Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. In: ICRA

Coumans E (2013) Bullet real-time physics simulation. URL http://bulletphysics.org

Delamarre Q, Faugeras OD (2001) 3d articulated models and multiview tracking with physical forces. CVIU 81(3):328–357

Ekvall S, Kragic D (2005) Grasp recognition for programming by demonstration. In: ICRA, pp 748–753

Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: A review. CVIU 108(1-2):52–73

Everingham M, Van Gool L, Williams C, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. IJCV 88(2):303–338

Felzenszwalb PF, Huttenlocher DP (2004) Distance transforms of sampled functions. Tech. rep., Cornell Computing and Information Science

Gall J, Fossati A, Van Gool L (2011a) Functional categorization of objects using real-time markerless motion capture. In: CVPR, pp 1969–1976

Gall J, Yao A, Razavi N, Van Gool L, Lempitsky V (2011b) Hough forests for object detection, tracking, and action recognition. PAMI 33(11):2188–2202

Gärtner B, Schönherr S (2000) An efficient, exact, and generic quadratic programming solver for geometric optimization. SCG '00, pp 110–118

Hamer H, Schindler K, Koller-Meier E, Van Gool L (2009) Tracking a hand manipulating an object. In: ICCV, pp 1475–1482

Hamer H, Gall J, Weise T, Van Gool L (2010) An object-dependent hand pose prior from sparse training data. In: CVPR, pp 671–678

Heap T, Hogg D (1996) Towards 3d hand tracking using a deformable model. In: FG, pp 140–145

Holzer S, Rusu R, Dixon M, Gedikli S, Navab N (2012) Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In: IROS, pp 2684–2689

Jones MJ, Rehg JM (2002) Statistical color models with application to skin detection. IJCV 46(1):81–96

Keskin C, Kra F, Kara Y, Akarun L (2012) Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In: ECCV

Kim D, Hilliges O, Izadi S, Butler AD, Chen J, Oikonomidis I, Olivier P (2012) Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In: UIST, pp 167–176

Kyriazis N, Argyros A (2013) Physically plausible 3d scene tracking: The single actor hypothesis. In: CVPR, pp 9–16

Kyriazis N, Argyros A (2014) Scalable 3d tracking of multiple interacting objects. In: CVPR

de La Gorce M, Fleet DJ, Paragios N (2011) Model-based 3d hand pose estimation from monocular video. PAMI 33(9):1793–1805

Lewis JP, Cordner M, Fong N (2000) Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In: SIGGRAPH

Lu S, Metaxas D, Samaras D, Oliensis J (2003) Using multiple cues for hand tracking and model refinement. In: CVPR, pp 443–450

MacCormick J, Isard M (2000) Partitioned sampling, articulated objects, and interface-quality hand tracking. In: ECCV, pp 3–19

Murray RM, Sastry SS, Zexiang L (1994) A Mathematical Introduction to Robotic Manipulation

Oikonomidis I, Kyriazis N, Argyros A (2011a) Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC, pp 101.1–101.11

Oikonomidis I, Kyriazis N, Argyros A (2011b) Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: ICCV

Oikonomidis I, Kyriazis N, Argyros AA (2012) Tracking the articulated motion of two strongly interacting hands. In: CVPR, pp 1862–1869

Oikonomidis I, Lourakis MI, Argyros AA (2014) Evolutionary quasi-random search for hand articulations tracking. In: CVPR

Paris S, Durand F (2009) A fast approximation of the bilateral filter using a signal processing approach. IJCV 81(1):24–52

Pons-Moll G, Rosenhahn B (2011) Model-Based Pose Estimation, pp 139–170

Qian C, Sun X, Wei Y, Tang X, Sun J (2014) Realtime and robust hand tracking from depth. In: CVPR

Rehg J, Kanade T (1995) Model-based tracking of self-occluding articulated objects. In: ICCV, pp 612–617

Rehg JM, Kanade T (1994) Visual tracking of high dof articulated structures: an application to human hand tracking. In: ECCV, pp 35–46

Romero J, Kjellström H, Kragic D (2009) Monocular real-time 3d articulated hand pose estimation. In: HUMANOIDS, pp 87–92

Romero J, Kjellström H, Kragic D (2010) Hands in action: real-time 3d reconstruction of hands in interaction with objects. In: ICRA, pp 458–463

Rosales R, Athitsos V, Sigal L, Sclaroff S (2001) 3d hand pose reconstruction using specialized mappings. In: ICCV, pp 378–387

Rosenhahn B, Brox T, Weickert J (2007) Three-dimensional shape knowledge for joint image segmentation and pose tracking. IJCV 73(3):243–262

Rusinkiewicz S, Levoy M (2001) Efficient variants of the icp algorithm. In: 3DIM, pp 145–152

Rusinkiewicz S, Hall-Holt O, Levoy M (2002) Real-time 3d model acquisition. TOG 21(3):438–446

Schmidt T, Newcombe R, Fox D (2014) Dart: Dense articulated real-time tracking. In: Proceedings of Robotics: Science and Systems, Berkeley, USA

Sharp T, Keskin C, Robertson D, Taylor J, Shotton J, Kim D, Rhemann C, Leichter I, Vinnikov A, Wei Y, Freedman D, Kohli P, Krupka E, Fitzgibbon A, Izadi S (2015) Accurate, robust, and flexible real-time hand tracking. In: CHI

Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR, pp 1297–1304

Sridhar S, Oulasvirta A, Theobalt C (2013) Interactive markerless articulated hand motion tracking using rgb and depth data. In: ICCV, pp 2456–2463

Sridhar S, Rhodin H, Seidel HP, Oulasvirta A, Theobalt C (2014) Real-time hand tracking using a sum of anisotropic gaussians model. In: 3DV

Sridhar S, Mueller F, Oulasvirta A, Theobalt C (2015) Fast and robust hand tracking using detection-guided optimization. In: CVPR

Stenger B, Mendonca P, Cipolla R (2001) Model-based 3D tracking of an articulated hand. In: CVPR

Stolfi J (1991) Oriented Projective Geometry: A Framework for Geometric Computation. Boston: Academic Press

Sudderth E, Mandel M, Freeman W, Willsky A (2004) Visual Hand Tracking Using Nonparametric Belief Propagation. In: Workshop on Generative Model Based Vision, pp 189–189

Tang D, Yu TH, Kim TK (2013) Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV, pp 3224–3231

Tang D, Chang HJ, Tejani A, Kim TK (2014) Latent regression forest: Structured estimation of 3d articulated hand posture. In: CVPR

Taylor J, Stebbing R, Ramakrishna V, Keskin C, Shotton J, Izadi S, Hertzmann A, Fitzgibbon A (2014) User-specific hand modeling from monocular depth sequences. CVPR

Teschner M, Kimmerle S, Heidelberger B, Zachmann G, Raghupathi L, Fuhrmann A, Cani MP, Faure F, Magnetat-Thalmann N, Strasser W (2004) Collision detection for deformable objects. In: Eurographics

Thayananthan A, Stenger B, Torr PHS, Cipolla R (2003) Shape context and chamfer matching in cluttered scenes. In: CVPR, pp 127–133

Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. TOG 33

Tzionas D, Gall J (2013) A comparison of directional distances for hand pose estimation. In: GCPR

Tzionas D, Srikantha A, Aponte P, Gall J (2014) Capturing hand motion with an rgb-d sensor, fusing a generative model with salient points. In: GCPR

Vaezi M, Nekouie MA (2011) 3d human hand posture reconstruction using a single 2d image. IJHCI 1(4):83–94

Wang RY, Popović J (2009) Real-time hand-tracking with a color glove. TOG 28(3):63:1–63:8

Wu Y, Lin J, Huang T (2001) Capturing natural hand articulation. In: ICCV, pp 426–432

Ye M, Zhang Q, Wang L, Zhu J, Yang R, Gall J (2013) A survey on human motion analysis from depth data. In: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, pp 149–187

(a) Fingers Walking

(b) Fingers Crossing

(c) Fingers Crossing and Twisting

(d) Fingers Dancing

(d) Fingers Hugging

(d) Fingers Grasping

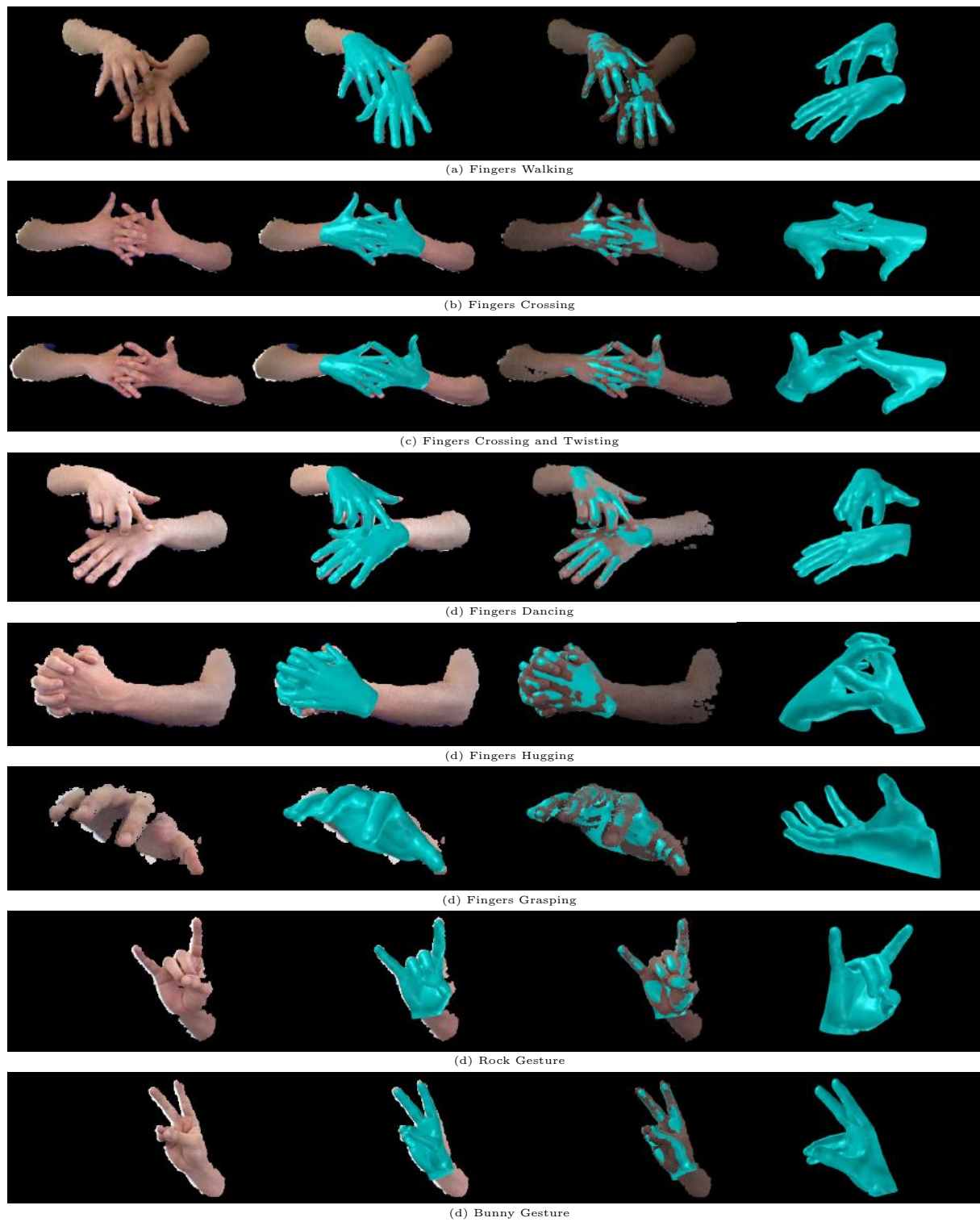(d) Rock Gesture

(d) Bunny Gesture

Fig. 18 Some of the obtained results. (Left) Input RGB-D image. (Center-Left) Obtained results overlayed on the input image. (Center-Right) Obtained results fitted in the input point cloud. (Right) Obtained results from another viewpoint.
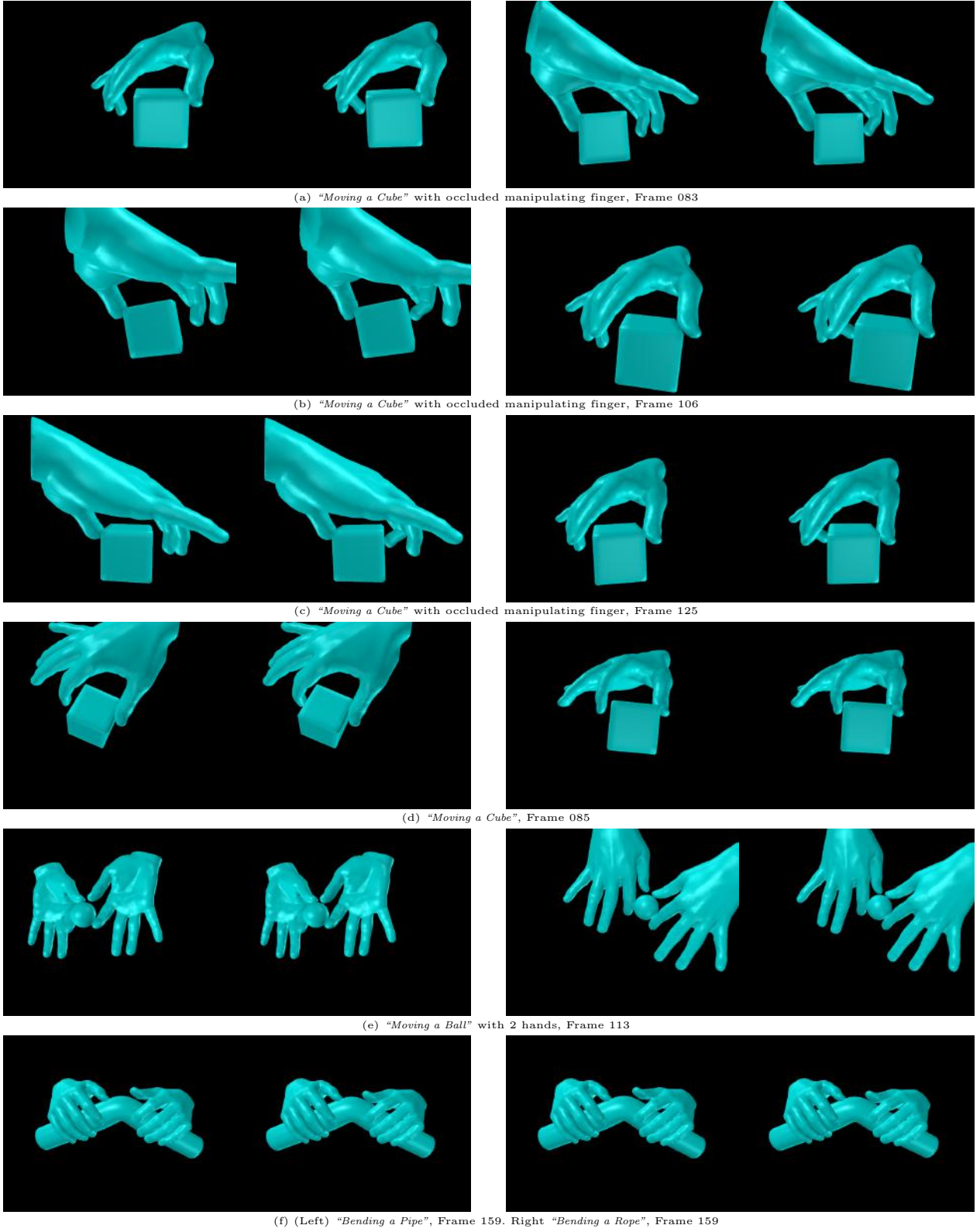
(a) *"Moving a Cube"* with occluded manipulating finger, Frame 083

(b) *"Moving a Cube"* with occluded manipulating finger, Frame 106

(c) *"Moving a Cube"* with occluded manipulating finger, Frame 125

(d) *"Moving a Cube"*, Frame 085

(e) *"Moving a Ball"* with 2 hands, Frame 113

(f) (Left) *"Bending a Pipe"*, Frame 159. Right *"Bending a Rope"*, Frame 159

Fig. 19 The impact of the physics component. For each image couple, the left image corresponds to LO + $\mathcal{SC}$x and the right one to LO + $\mathcal{SCP}$. In the case of missing or ambiguous input visual data, as in sequences with occluded manipulating finger, the contribution of the physics component towards better physically plausible poses becomes more prominent
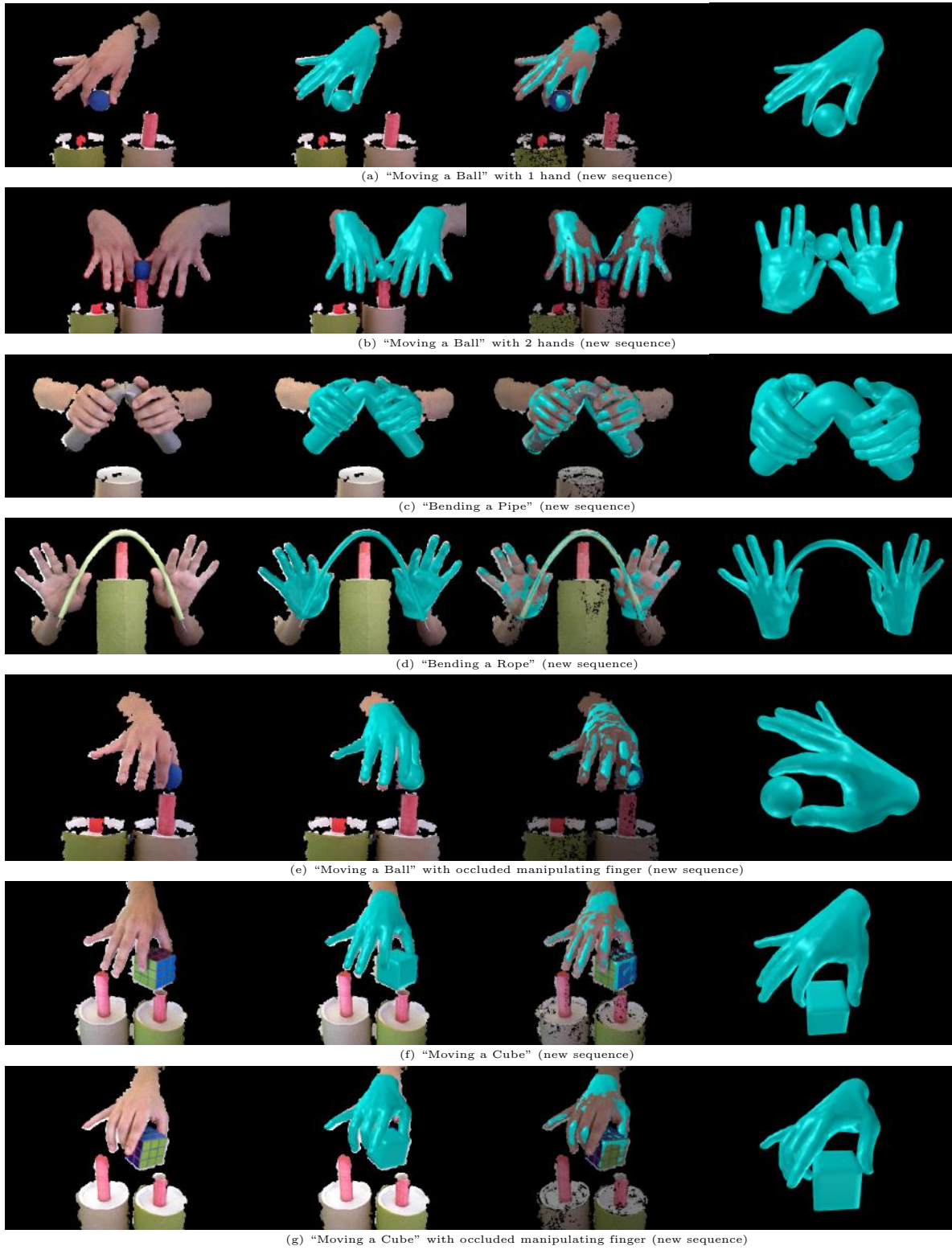
(a) "Moving a Ball" with 1 hand (new sequence)

(b) "Moving a Ball" with 2 hands (new sequence)

(c) "Bending a Pipe" (new sequence)

(d) "Bending a Rope" (new sequence)

(e) "Moving a Ball" with occluded manipulating finger (new sequence)

(f) "Moving a Cube" (new sequence)

(g) "Moving a Cube" with occluded manipulating finger (new sequence)

Fig. 20 Some of the obtained results. (Left) Input RGB-D image. (Center-Left) Obtained results overlayed on the input image. (Center-Right) Obtained results fitted in the input point cloud. (Right) Obtained results from another viewpoint.
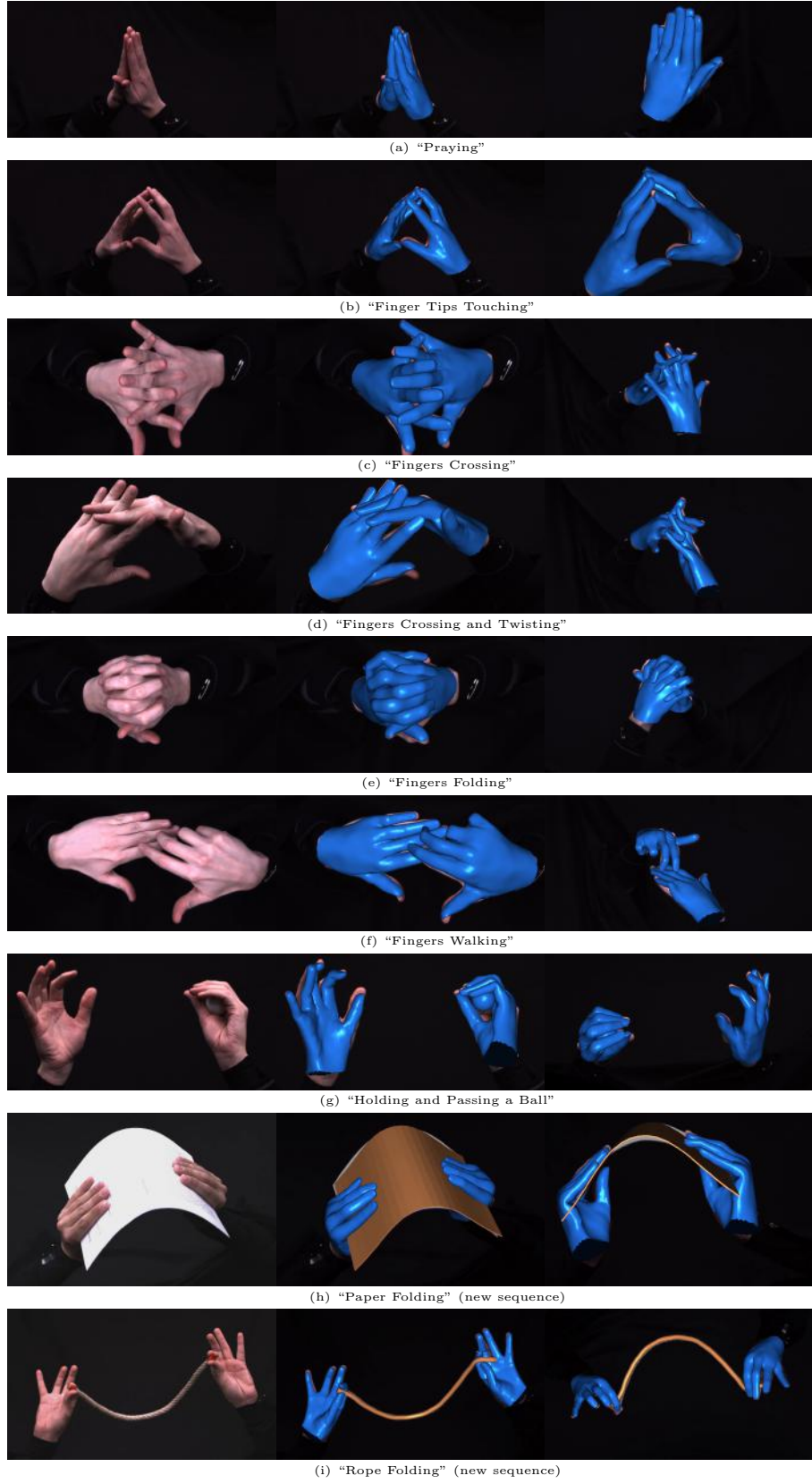
(a) "Praying"

(b) "Finger Tips Touching"

(c) "Fingers Crossing"

(d) "Fingers Crossing and Twisting"

(e) "Fingers Folding"

(f) "Fingers Walking"

(g) "Holding and Passing a Ball"

(h) "Paper Folding" (new sequence)

(i) "Rope Folding" (new sequence)

Fig. 21 Some of the obtained results. (Left) One of the input RGB images. (Center) Obtained results overlayed on the input image. (Right) Obtained results from another viewpoint.