Open access • Proceedings Article • DOI:10.1109/CVPR.2012.6247885

Tracking the articulated motion of two strongly interacting hands — Source link

Iason Oikonomidis, Nikolaos Kyriazis, Antonis A. Argyros

Institutions: University of Crete

Published on: 16 Jun 2012 - Computer Vision and Pattern Recognition

Topics: Optimization problem, Stochastic optimization, Particle swarm optimization, Match moving and Video tracking

Tracking the Articulated Motion of Two Strongly Interacting Hands

I. Oikonomidis, N. Kyriazis, A.A. Argyros
Institute of Computer Science, FORTH and Computer Science Deparment, University of Crete
{oikonom, kyriazis, argyros}@ics.forth.gr

Abstract

We propose a method that relies on markerless visual observations to track the full articulation of two hands that interact with each-other in a complex, unconstrained manner. We formulate this as an optimization problem whose 54*-dimensional parameter space represents all possible configurations of two hands, each represented as a kinematic structure with* 26 *Degrees of Freedom (DoFs). To solve this problem, we employ Particle Swarm Optimization (PSO), an evolutionary, stochastic optimization method with the objective of finding the two-hands configuration that best explains observations provided by an RGB-D sensor. To the best of our knowledge, the proposed method is the first to attempt and achieve the articulated motion tracking of two strongly interacting hands. Extensive quantitative and qualitative experiments with simulated and real world image sequences demonstrate that an accurate and efficient solution of this problem is indeed feasible.*

1. Introduction

The problem of tracking the articulation of the human body from markerless visual observations is of both theoretical interest and practical importance. From a theoretical point of view, the problem is intriguing since humans solve it effortlessly and effectively. From an application-oriented perspective, a solution to this problem facilitates non-intrusive human motion capture and constitutes a fundamental building block towards human activity recognition, human-computer interaction, robot learning by demonstration, etc.

Despite the significant progress in the last years, the problem remains unsolved in its full extent [12]. Difficulties stem from the high dimensionality of the configuration space of the human body, the varying appearance of humans and the self-occlusions of human parts.

In this work we are particularly interested in the problem of tracking hand articulations. A point in a 26-dimensional configuration space defines the global position and orientation of the hand plus the 20 joint angles between various hand parts. Because of its flexibility that generally induces a concave shape, a performing hand is severely self occluded even when observed from purposefully selected viewpoints. Thus, the markerless tracking of a hand constitutes a high dimensional search problem that needs to be solved based on incomplete and possibly ambiguous observations.

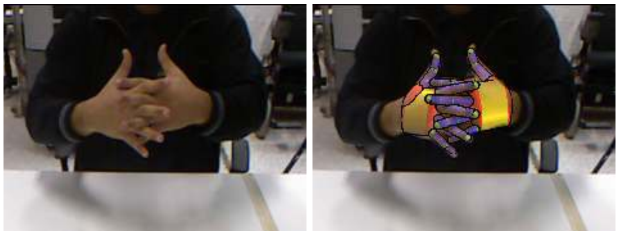

Figure 1. Left: A view of two interacting hands. Right: The configuration of the two hands as estimated by the proposed method, superimposed on the left frame (cropped 320×240 regions from the original 640×480 images).

Tracking two hands in interaction with each other is an even more interesting problem. The interest stems from the fact that a plethora of human activities (object grasping and manipulation, sign language, social interaction) involve collaborative use and strong interaction of both hands. Consider, as an example, the situation shown in Fig. 1. For a human observer, the interpretation of the joint configuration of the two hands is immediate. Even more interestingly, this interpretation is associated to the joint hand configuration rather than to each individual hand. Thus, the availability of computational techniques that are able to jointly infer the full articulation of the hands in such scenarios, opens new avenues in the interpretation of human activities.

Compared to the already difficult problem of tracking the articulation of a single hand, the problem of tracking two hands is even more challenging. If the two hands are clearly separated in the field of view of the observer, it would suffice to solve two instances of the single-hand tracking problem. However, if hands interact with each other, the situation becomes much more complicated. Besides the self-occlusions of each individual hand, further occlusions are introduced because of the complex inter-relations of the two hands, each hiding important observations of the other. Even further, the available, fewer observations become more ambiguous since the existence of parts from two hands increase the number of potential interpretations. Essentially, each hand acts as a distractor to the interpretation of the other.

The direct implication of the above observations is that it is very difficult for any tracker of a single hand to cope effectively with the problem of tracking two interacting hands. The configuration of each hand can only be inferred in the context of its interaction with the other. This calls for a holistic approach, in which a joint model of the two interacting hands is considered. In such a framework, the desired outcome is the two-hands configuration that not only best explains all available observations, but also explains the ones that are missing due to the hands interaction.

In this paper we follow this approach. We consider a model of two hands, potentially in strong interaction, and we formulate an optimization problem whose solution is the position, pose and full articulation of two hands that best explain the set of all available visual observations. We also demonstrate that despite its large dimensionality, this problem can be solved both effectively and efficiently.

1.1. Related work

To the best of our knowledge, there is no existing work that addresses the problem of tracking the full articulation of two interacting hands from markerless visual observations. We therefore provide an overview of works on single-hand articulation tracking and discuss their potential extensibility to the problem of tracking two interacting hands.

Hand pose estimation and tracking methods can be categorized into appearance- and model-based ones [6]. *Appearance-based methods* employ an offline training process for establishing a mapping from a set of image features to a finite set of hand model configurations [3, 18–20, 23]. The discriminative power of these methods depends on the invariance properties of the employed features, the number and the diversity of the training postures and the method used to derive the mapping. Appearance-based methods are appropriate for recognizing a small set of known and diverse target hand configurations and less suitable in situations where accurate pose estimation of a freely performing hand is required. Scenarios involving two hands seem challenging for such methods. This is because the offline training process should consider the combinatorial space of the configurations of the two hands as well as the change in appearance of these configurations because of the different viewpoints of observation.

Model-based approaches [5,7,13–15,17,21,22] generate hand model hypotheses and evaluate them on the available visual observations. An optimization problem is formulated, whose objective function measures the discrepancy between observed and synthesized visual cues that are generated based on a specific hand posture hypotheses. The employed optimization method should be able to evaluate the objective function at arbitrary points in the multidimensional model parameters space. Thus, unlike appearance-based methods, most of the computations need to be performed online. On the positive side, such methods avoid the time and effort consuming task of training and they provide continuous solutions to the problem of hand pose recovery.

Another categorization is based on how partial evidence regarding the individual rigid parts of the articulated object contributes to the final solution [14]. *Disjoint evidence methods* [7, 17, 20, 22] consider individual parts in isolation prior to evaluating them against observations. *Joint evidence methods* [3, 5, 13–15, 18, 19, 21, 23] consider all parts in the context of complete articulated object hypotheses. By construction, joint-evidence methods treat part interactions effortlessly, but their computational requirements are rather high. Disjoint evidence methods usually have lower computational requirements than joint-evidence ones, but need to explicitly handle part interactions such as collisions and occlusions. Since such issues are pronounced in the problem of two hands tracking, joint evidence methods are apparently more suitable than disjoint evidence methods.

1.2. Contribution

In terms of the previously described classifications, this paper presents a model-based, joint-evidence method for tracking the full articulation of two interacting hands. Observations come from an off-the-shelf RGB-D sensor (Kinect [11]). Two-hands tracking is formulated as an optimization problem. The objective function to be minimized quantifies the discrepancy between the 3D structure and appearance of hypothesized configurations of two hands and the corresponding visual observations. Optimization is performed through a variant of an evolutionary optimization method (Particle Swarm Optimization - PSO) tailored to the needs of the specific problem.

From a methodological point of view, the proposed approach combines the merits of two recently proposed methods for tracking hand articulations [14, 15]. More specifically, in [14], we proposed a joint-evidence method for tracking the full articulation of a single, isolated hand based on data provided by a Kinect. We extend this approach so that it can track two strongly interacting hands.

Our method is also related to the one presented in [15] that tracks a hand interacting with a known rigid object. The fundamental idea behind that work is to model hand-object relations and to treat occlusions as a source of information rather than as a complicating factor. We extend this idea by demonstrating that it can be exploited effectively in solving the much more complex problem of tracking two articulated objects (two hands). Additionally, this more complex problem is solved based on input provided by a compact

Kinect sensor, as opposed to the multicamera calibrated system employed in [15].

Experimental results demonstrate that the accuracy achieved in two hands tracking is in the order of 6mm, in scenarios involving very complex interaction between two hands. Interestingly, despite the large increase in the dimensionality of the problem compared to [14] (from 27 to 54 problem dimensions), the computational budget required for achieving this accuracy is only slightly increased.

The major contributions of this work can be summarized as follows: (a) We present the first method for accurate, robust and efficient tracking of the articulated motion of two hands in strong interaction, a problem that has never been addressed before. (b) We demonstrate that the core method presented in [14] can be naturally extended to handle the problem of tracking the articulation of two interacting hands. (c) We demonstrate that the idea of modeling context and occlusions as presented in [15] can be exploited towards tracking the articulation of two interacting hands. (d) We demonstrate that despite the doubling of the dimensionality of the problem compared to [14], the proposed approach achieves comparable accuracy with a comparable computational budget.

2. Tracking two interacting hands

The proposed method achieves tracking of two interacting hands by directly attributing sensory information to the joint articulation of two synthetic and symmetric 3D hand models, of known size and kinematics (see Fig. 2). For given articulations of two hands we are able to predict what the RGB-D sensor would perceive, by simulating the acquisition process, i.e. producing synthetic depth maps for specific camera-scene calibrations. Having established a parametric process that produces comparable data to the actual input, we perform tracking by searching for the parameters that produce depth maps which are most similar to the actual input.

Tracking is performed in an online fashion, where at each step and for every new input an optimization problem is solved. A variant of the PSO search heuristic is used to minimize the discrepancy between the actual RGB-D input and simulated depth maps, generated from hypothesized articulations. The best scoring hypothesis constitutes the solution for the current input. The discrepancy measure is carefully formulated so that robustness is achieved. Towards computational efficiency, temporal continuity is exploited at each optimization step.

2.1. Input/preprocessing

The input from the RGB-D sensor [16] consists of an RGB image I and a corresponding depth map D, i.e. a depth value for every pixel in I. The dimensions of both arrays are 640×480. A skin color map o_s is produced from I, by means of [2]. From o_s and D a new depth map o_d is computed, where only depth values of D that correspond to skin colored pixels in o_s are kept.

2.2. Model/search space

We define a parametric model of the joint kinematics of two hands. As already discussed, it is of vital importance to consider both hands cojointly, so that we can effectively perform inference over their potentially strong interaction. The parametric model of the two hands coincides with the search space of each optimization step. Each of the hands has 27 parameters, that represent the hand's pose (3-D position and 4-D quaternion-encoded orientation) and 4-D articulations of each of the 5 fingers (a 2-D revolute joint that connects the palm with the finger and two 1-D revolute joints that connect adjacent phallanges). The ranges of parameter values are linearly bounded, according to anatomical studies [1]. For two hands the dimensionality of the search space amounts to twice the dimensionality for one hand (i.e. 54), as we do not consider any additional constraints over their joint motion.

2.3. Simulation/comparable features

For each point h in the search space (see Sec. 2.2) a mapping to the feature space of the actual observations is required. We simulate the acquisition process of the depth sensor by means of rendering. Each point h defines two 3D skeletons by applying forward kinematics over the parameters detailed in Sec. 2.2. These skeletons are skinned with appropriately transformed instances of 3D spheres and cylinders. The usage of only two primitives proves to be computationally efficient (see Sec. 2.7). Given the calibration information C for the RGB-D camera, we rasterize a depth map $r_d(h, C)$ from the implicit 3D structure described so far. The resulting model is very similar to the one we used in [14]. The rendered 3D structure is depicted in Fig. 2(d).

2.4. Discrepancy/objective function

The objective function to be optimized is essentially a penalty function to be minimized. This penalty is defined with respect to a tracking frame's observation O and a rendered depth map $r_d(h, C)$ that is generated from a hypothesis h. The penalty function $E(\cdot)$ consists of two terms, a prior term $P(\cdot)$ and a data term $D(\cdot)$:

$$E(O, h, C) = P(h) + \lambda_k \cdot D(O, h, C), \qquad (1)$$

where $\lambda_k = 2$ is a regularization parameter.

The box bounds of the search space are not expressive enough to tightly define the region of valid hand articulations. Within these bounds, $P(\cdot)$ penalizes invalid articulation hypotheses. In this work we invalidate articulations

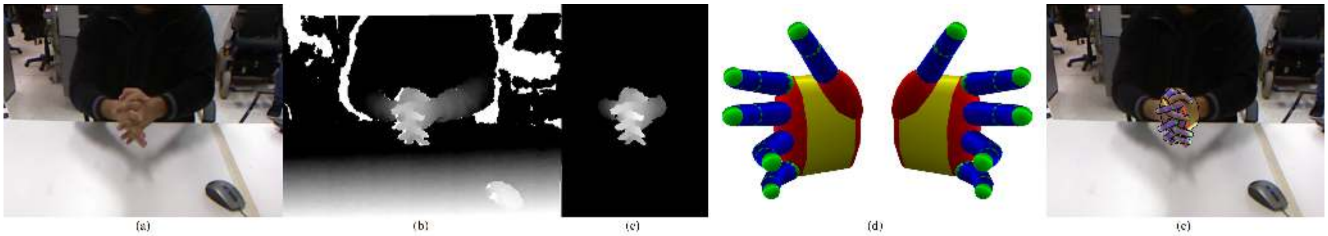

Figure 2. By masking the depth information (b), with a skin color detection performed upon RGB data (a), a depth map (c) of image regions corresponding to hands is extracted, from Kinect input. The proposed method fits the 54-D joint model of two hands (d) onto these observations, thus recovering the hand articulation that best explains the observations (e).

where adjacent fingers inter-penetrate. Thus,

$$P(h) = \sum_{p \in \mathrm{Q}} - \min(\phi(p, h), 0), \tag{2}$$

where Q amounts to the pairs of adjacent fingers, and ϕ is the abduction-adduction difference (in rads) of adjacent fingers, excluding the thumb. We have indeed tried elaborate and more computationally expensive collision models to penalize inter-penetration but we have found simple angle differences to efficiently resolve challenging tracking scenarios.

The term $D(\cdot)$ quantifies the incompatibility of input O to an articulation hypothesis h. Essentially, it is the result of the comparison of two parts. The first part $O = \{o_s, o_d\}$ consists of the input depth map o_d and the skin map o_s. The other part, referring to a hypothesis h, consists of a simulated depth map $r_d(h, C)$ and an implicitly defined skin map $r_s(h, C)$, that is set at points where $r_d(h, C)$ is occupied. The main purpose of $D(\cdot)$ is to penalize depth discrepancies. However, to make it robust, a few more points need to be addressed.

Unless depth differences are clamped within a predetermined range d_M, large differences, that can be due to noise, dominate and produce a false high penalty. By clamping we make $D(\cdot)$ smoother and, thus, add noise tolerance in its optimization. Moreover, we also consult the overlap of the actual and simulated skin maps. More specifically, hypotheses resulting in significant overlap with actual skin maps are preferred even if they result in slightly greater depth discrepancies. Empirical evaluation has proven that this approach eliminates strong local minima around the global minimum and therefore facilitates the convergence of the optimization process to its true optimum.

The aforementioned are encoded in the following penalty function:

$$\begin{aligned} D(O, h, C) \quad &= \quad \lambda \frac{\sum min(|o_d - r_d|, d_M)}{\sum(o_s \vee r_s) + \epsilon} + \\ &\left(1 - \frac{2\sum(o_s \wedge r_s)}{\sum(o_s \wedge r_s) + \sum(o_s \vee r_s)}\right), \end{aligned} \tag{3}$$

where $\lambda = 0.05$ acts as a regularization parameter, d_M is set to $4cm$ and $\epsilon = 10^{-6}$ is added to denominators in order to avoid possible divisions by zero. Differences are normalized over their effective areas.

2.5. Search/optimization

The challenging task of optimization at each tracking frame is delegated to the powerful Particle Swarm Optimization (PSO) search heuristic [8, 9]. PSO is an evolutionary optimization algorithm that receives an objective function $F(\cdot)$ and a search space S and outputs an approximation of the optimum of $F(\cdot)$ in S, while treating it as a black box. Being evolutionary, it is parameterized with respect to a population of particles. These parameters amount to the particle count N and the generation count G. Three additional parameters, namely w (constriction factor [4]), c_1 (cognitive component) and c_2 (social component), adjust the behavior of the algorithm.

For each generation k and particle i PSO maintains a state that consists of a global optimum position G_k, a local optimum $P_{k,i}$, the current position $x_{k,i}$ and the current velocity $v_{k,i}$. Initially, particles are sampled uniformly in S. At each generation, the velocity of each particle is updated according to

$$v_{k+1,i} = w(v_{k,i} + c_1 r_1 (P_{k,i} - x_{k,i}) + c_2 r_2 (G_k - x_{k,i})) \tag{4}$$

and the current position of each particle is updated according to:

$$x_{k+1,i} = x_{k,i} + v_{k+1,i}. \tag{5}$$

$P_{k+1,i}$ is set to

$$P_{k+1,i} = \begin{cases} x_{k+1,i}, & F(x_{k+1,i}) < F(P_{k+1,i}) \\ P_{k,i}, & \text{otherwise} \end{cases} \tag{6}$$

G_k is set to the best scoring particle's $P_{k,i}$:

$$G_{k+1} = P_{k+1,l}, \text{ with } l = \arg\min_m \left(F\left(P_{k+1,m}\right)\right). \tag{7}$$

Variables r_1, r_2 represent uniformly distributed random numbers in the range $[0, 1]$.

As suggested in [4], we fix the behavioral parameters to $c_1 = 2.8$, $c_2 = 1.3$ and

$$w = 2 / \left| 2 - \psi - \sqrt{\psi^2 - 4\psi} \right| \tag{8}$$

with $\psi = c_1 + c_2$. We have experimentally confirmed that for the w as defined in Eq.(8), any combination that satisfies $c_1 + c_2 = 4.1$ achieves essentially the same optimization performance.

There are traits that make PSO attractive to use in a tracking method. It is derivative-agnostic, which makes it easy to try and optimize arbitrary objective functions, with no limitations over convexity, continuity etc. Moreover, its performance depends on essentially two parameters, namely N and G.

In the proposed method a variant of PSO is considered that better suits our tracking requirements. As already stated in [14], the original PSO algorithm has been effective in accurately recovering the pose of the hand's palm (6 DoFs). However, less accuracy has been observed for the fingers (the rest of the 20 DoFs). This occurs due to premature "collapsing" [9] of the population. In order to alleviate this, we employ additional randomization over these remaining parameters, so that their range is better explored [24]. This process is applied to the joint parameter space of both hands (54 DoFs) and for the 40 parameters that regard the 10 fingers.

Additionally, we exploit the parallel nature of PSO by delegating evaluations of individual particles to distinct computational cores of a parallel platform. Each generation is evaluated in parallel, given that the score of each particle is independent to any other. This introduces significant benefits with respect to execution times.

2.6. Tracking loop

In order to perform tracking across time we iterate over instances of the same optimization problem. Each iteration is performed on new input provided from the sensor and yields a new pose estimate. In order to provide such an estimate, $E(\cdot)$ is minimized by PSO. What differs from frame to frame is the input and the effective search area that is provided to PSO.

For every new frame, the newly acquired input is preprocessed and mapped into the feature space of skin and depth measurements, as variable O. All subsequent evaluations of $E(\cdot)$ are performed based on this input. Every hypothesis h that is generated by PSO is rendered and thus mapped to the same feature space. PSO drives an exploratory course in which multiple invocations ($N \times G$) of $E(O, h, C)$ are made. The optimal hypothesis

$$h_{max} = \arg\min_{h} E(O, h, C) \tag{9}$$

is output as the inferred articulation for the current tracking frame.

Although the original PSO requires a uniform initialization of its population in S, we exploit temporal continuity and constrain the effective search area. To do so, for every next tracking frame we initialize the population to be in the vicinity of h_{max} of the previous frame. The optimum h_{max} of the previous frame is copied to the new population. The rest of the population consists of random perturbations of h_{max}. This strategy makes our proposed method suitable for tracking but has the disadvantage that the hand poses must be initialized for the first observed frame of a sequence.

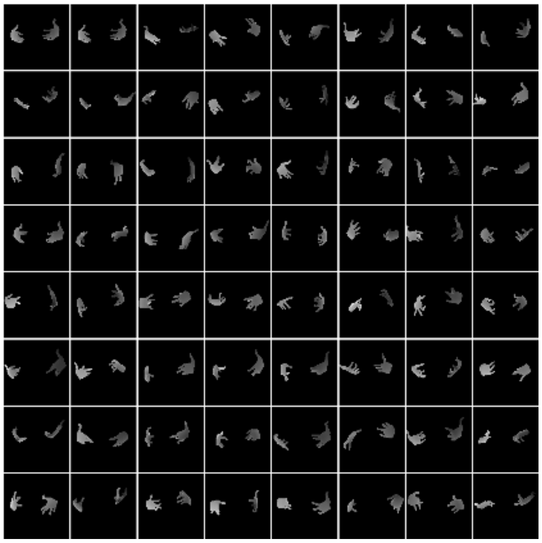

Figure 3. Feature mapping of an entire generation of model hypotheses that can be generated and evaluated in sub-millisecond time scale on a GPU.

2.7. Parallel implementation

The execution time of the presented tracking loop is dominated by the evaluation of the data term $D(\cdot)$ of the penalty function $E(\cdot)$ (see Eq. (1)). 3D rendering and operations over entire maps induce costs that are prohibitive for mainstream CPUs but can be efficiently handled by contemporary GPUs. We exploit parallelism by considering renderings of multiple hypotheses, simultaneously, in big tiled renderings. Essentially, an entire generation is feature-mapped upon a single 2D array, as shown in Fig.3. Per pixel computations are implemented using shaders and the required summations are performed by means of *mip* (multum in parvo) mapping with the addition operator. Following the guidelines of [10], we employ hardware instancing and multi-viewport clipping in order to efficiently cope with many model hypotheses that consist of homogeneous transformations of just a sphere and a cylinder.

3. Experimental evaluation

Synthetic data as well as real-world sequences obtained by a Kinect sensor [11] were used to experimentally evaluate the proposed method. Experiments were performed on a computer equipped with a quad-core Intel i7 950 CPU, 6

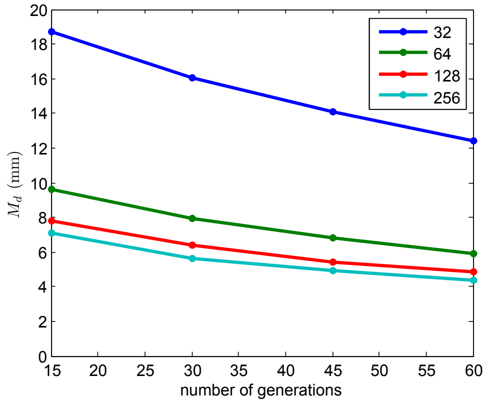

Figure 4. Quantitative evaluation of the performance of the method with respect to the PSO parameters. Each line of the graph corresponds to a different number of particles as shown in the legend.

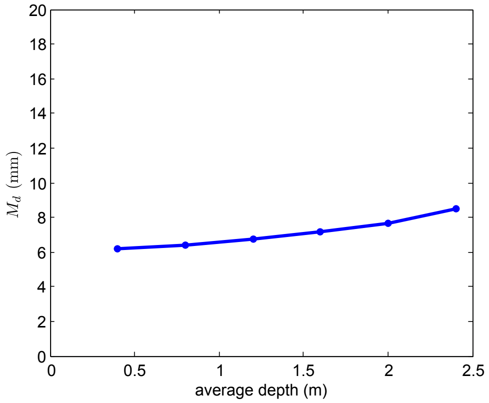

Figure 5. Quantitative evaluation of the performance of the method with respect to the average distance from the sensor.

GB RAM and an Nvidia GTX 580 GPU with 1581 $GFlops$ processing power and 1.5 GB of memory.

3.1. Experiments on synthetic data

The quantitative evaluation of the proposed method has been performed using synthetic data. This approach is often encountered in the relevant literature [7, 13–15] because ground truth data for real-world image sequences is hard to obtain. The employed synthetic sequence consists of 300 poses that encode typical interactions of two hands. Rendering was used to synthesize the required input O. To quantify the accuracy in hand pose estimation, we adopt the metric used in [7]. More specifically, the distance between corresponding phalanx endpoints in the ground truth and in the estimated hand poses is measured. The average of all these distances, for both hands, over all the frames of the sequence constitutes the resulting error estimate Δ. It is worth noting that these distances include estimations for hand points that, because of occlusions, are not observable.

The influence of several factors to the performance of the method was assessed in respective experiments. Figure 4 illustrates the behavior of the method with respect to the PSO parameters (number of generations and particles per generation). The product of these parameters determines the computational budget of the proposed methodology, i.e. the number of objective function evaluations for each each tracking frame. The horizontal axis of the plot denotes the number of PSO generations. Each plot of the graph corresponds to a different number of particles per generation. Each point in each plot is the median M_d of the error Δ for 20 repetitions of an experiment run with the specific parameters. A first observation is that M_d decreases monotonically as the number of generations increase. Additionally, as the particles per generation increase, the resulting error decreases. Nevertheless, employing more that 45 generations and more than 64 particles results in disproportionally small improvement of the method's accuracy. The gains are at most $2mm$ or roughly 30%, for a 5-fold increase in computational budget. For this reason, the configuration of 64 particles for 45 generations was retained in all further experiments. In terms of computational performance, tracking is achieved at a framerate of 4Hz on the computational infrastructure described in Sec.3.

In another experiment we assessed the effect of varying the distance of the hands from the hypothesized sensor. By doing so, we explored the usefulness of the method in different application scenarios that require observations of a certain scene at different scales (e.g., close-up views of hands versus distant views of a human and his/her broader environment). To do this, we generated the same synthetic sequences at different average depths. The results of this experiment are presented in Fig. 5. At a distance of 50cm the error is equal to $6mm$. As the distance increases, the error also increases; Interestingly though, it doesn't exceed $8.5mm$ even at an average distance of $2.5m$. The used synthetic maps do not contain any kind of noise, in contrast to what happens in practice: the amount of noise is related to the distance from the sensor for data acquired with a Kinect.

The tolerance of the method to noisy observations was also evaluated. Two types of noise were considered. Errors in depth estimation were modeled as a Gaussian distribution centered around the actual depth value with the variance controlling the amount of noise. Skin-color segmentation errors were treated similarly to [18], by randomly flipping the label (skin/non-skin) of a percentage of pixels in the synthetic skin mask. Figure 6 plots the method's error in hand pose estimation for different levels of depth and skin segmentation error. As it can be verified, the hand pose recovery error is bounded in the range $[6mm..23mm]$, even in data sets very heavily contaminated with noise.

The accuracy in hand pose estimation with respect to

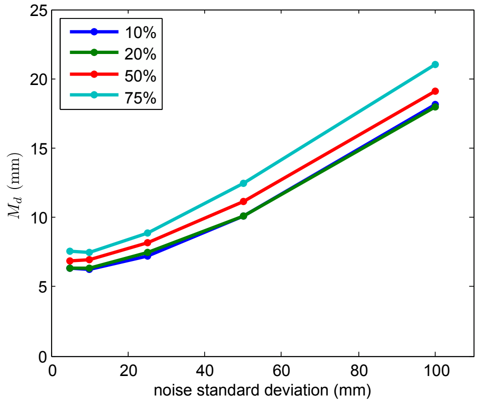

Figure 6. Quantitative evaluation of the performance of the method with respect to synthesized depth and skin-color detection noise.

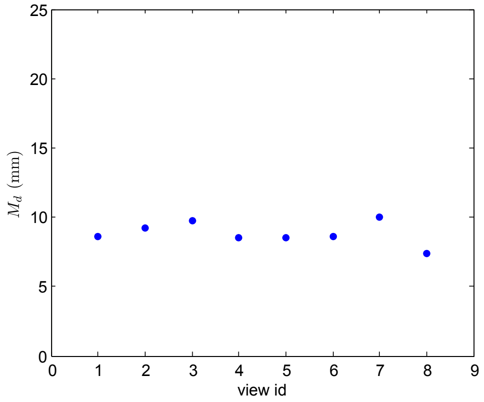

Figure 7. Quantitative evaluation of the performance of the method with respect to viewpoint variation.

viewpoint variations was also assessed. This was achieved by placing the virtual camera at 8 positions dispersed on the surface of a hemisphere around the hypothesized scene. The data points of Fig. 7 demonstrate that viewpoint variations do not significantly affect the performance of the method.

In a final experiment, we measured the performance of our single hand tracker [14] on the synthetic data set of the previous experiments. To do so, the system described in that work was used to track one of the two visible hands. The resulting error M_d for this experiment was 145mm. In practice, the single hand tracker is able to track accurately one of the two hands while it is not in interaction with the other. However, as soon as occlusions become extended due to hands interaction (for example, when one hand passes in front of the other), the track is often completely lost.

3.2. Experiments on real world sequences

Towards the qualitative evaluation of the proposed approach in real data, several long real-world image sequences were captured using the PrimeSense Sensor Module of OpenNI [16]. The supplemental material accompanying the paper provides a video with the results obtained from one such sequence (1776 frames)[1]. Indicative snapshots are shown in Fig. 8. Evidently, the estimated hand postures are in very close agreement with the image data, despite the complex articulation and strong interactions of the two hands.

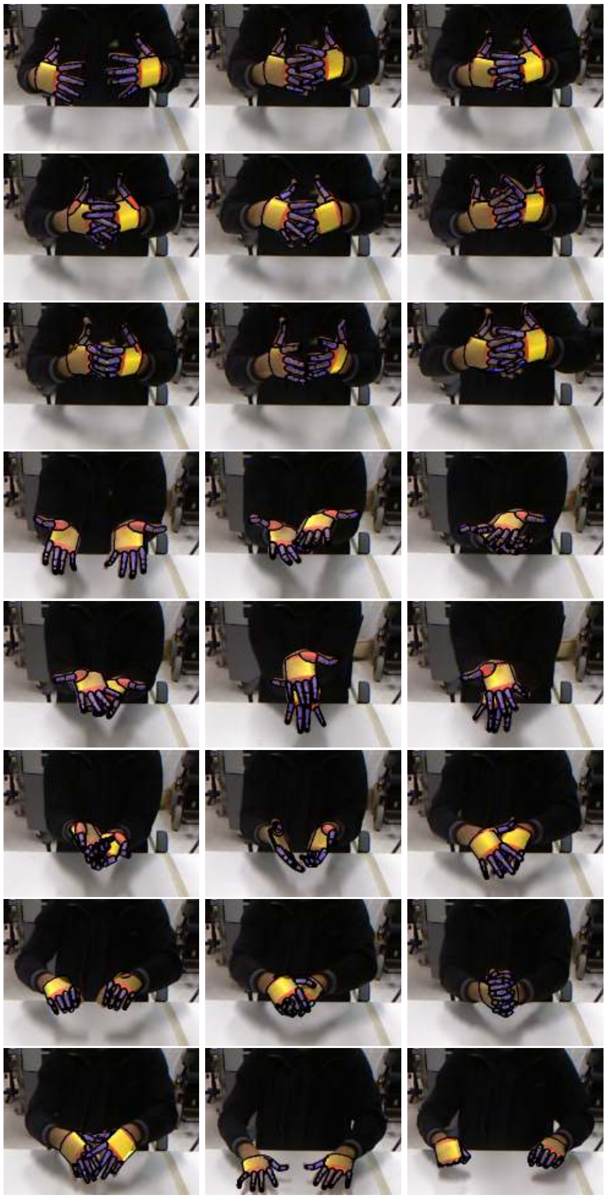

Figure 8. Snapshots from an experiment where two hands interact with each other (cropped 320 × 240 regions from the original 640 × 480 images).

[1] Available online at http://youtu.be/e3G9soCdIbc

4. Discussion

We proposed a method for tracking the full articulation of two strongly interacting hands, based on observations acquired by an RGB-D sensor. The problem was formulated as an optimization problem in a 54-dimensional parameter space spanning all possible configurations of two hands. Optimization seeks for the joint hand configuration that minimizes the discrepancy between rendered hand hypotheses and actual visual observations. Particle Swarm Optimization proved to be competent in solving this high dimensional optimization problem. More specifically, extensive experimental results demonstrated that accurate and robust tracking of two interacting hands can be achieved with an accuracy of 6mm at a framerate of 4Hz. Experimental results also demonstrated that in the presence of strong hand interactions, the straightforward alternative of solving two instances of a single hand tracking problem results in a much lower accuracy.

The proposed approach is the first to achieve a solution to this interesting and challenging problem. Hopefully, it will constitute an important building block in a large spectrum of application domains that critically depend on the accurate markerless perception of bi-manual human activities.

Acknowledgments

This work was partially supported by the IST-FP7-IP-215821 project GRASP and the IST-FP7-IP-288533 project RoboHow.Cog. The contribution of Pashalis Padeleris and Konstantinos Tzevanidis, members of the CVRL labolatory is gratefully acknowledged.

References

[1] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and Animation of Anatomically Based Human Hand Models. In *Eurographics symposium on Computer animation*, page 109. Eurographics Association, 2003. 3

[2] A. A. Argyros and M. Lourakis. Real-time Tracking of Multiple Skin-colored Objects with a Possibly Moving Camera. In *ECCV*, pages 368–379. Springer, 2004. 3

[3] V. Athitsos and S. Sclaroff. Estimating 3D Hand Pose From a Cluttered Image. In *CVPR*, pages II–432–9. IEEE, 2003. 2

[4] M. Clerc and J. Kennedy. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *Transactions on Evolutionary Computation*, 6(1):58–73, 2002. 4

[5] M. De La Gorce, N. Paragios, and D. J. Fleet. Model-based Hand Tracking With Texture, Shading and Self-occlusions. In *CVPR*, pages 1–8. IEEE, Jun. 2008. 2

[6] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based Hand Pose Estimation: A review. *CVIU*, 108(1-2):52–73, 2007. 2

[7] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a Hand Manipulating an Object. In *ICCV*, 2009. 2, 5

[8] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, Jan. 1995. 4

[9] J. Kennedy, R. Eberhart, and S. Yuhui. *Swarm intelligence*. Morgan Kaufmann, 2001. 4, 5

[10] N. Kyriazis, I. Oikonomidis, and A. A. Argyros. A GPU-powered Computational Framework for Efficient 3D Model-based Vision. Technical Report TR420, ICS-FORTH, Jul. 2011. 5

[11] Microsoft Corp. Redmond WA. Kinect for Xbox 360. 2, 5

[12] T. B. Moeslund, A. Hilton, and V. Kru. A Survey of Advances in Vision-based Human Motion Capture and Analysis. *CVIU*, 104:90–126, 2006. 1

[13] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Markerless and Efficient 26-DOF Hand Pose Recovery. In *ACCV*, pages 744–757. Springer, 2010. 2, 5

[14] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. In *BMVC*, Dundee, UK, Aug. 2011. 2, 3, 5, 6

[15] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints. In *ICCV*, pages 2088–2095. IEEE, Nov. 2011. 2, 3, 5

[16] OpenNI. PrimeSense Sensor Module, 2011. 3, 6

[17] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617. IEEE, 1995. 2

[18] J. Romero, H. Kjellström, and D. Kragic. Monocular Real-time 3D Articulated Hand Pose Estimation. In *International Conference on Humanoid Robots*, pages 87–92. IEEE, Dec. 2009. 2, 6

[19] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3D Hand Pose Reconstruction Using Specialized Mappings. *ICCV*, pages 378–385, 2001. 2

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. In *CVPR*. IEEE, 2011. 2

[21] B. Stenger, P. Mendonça, and R. Cipolla. Model-based 3D Tracking of an Articulated Hand. In *CVPR*, pages II–310–II–315. IEEE, 2001. 2

[22] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual Hand Tracking Using Nonparametric Belief Propagation. In *CVPR Workshop*, pages 189–189, 2004. 2

[23] Y. Wu and T. S. Huang. View-independent Recognition of Hand Postures. In *CVPR*, volume 2, pages 88–94. IEEE, 2000. 2

[24] T. Yasuda, K. Ohkura, and Y. Matsumura. Extended PSO with Partial Randomization for Large Scale Multimodal Problems. In *World Automation Congress*, pages 1–6. IEEE, Apr. 2010. 5