Scalable 3D Tracking of Multiple Interacting Objects

Nikolaos Kyriazis, Antonis Argyros
Institute of Computer Science, FORTH and Computer Science Department, University of Crete
{kyriazis, argyros}@ics.forth.gr

Abstract

We consider the problem of tracking multiple interacting objects in 3D, using RGBD input and by considering a hypothesize-and-test approach. Due to their interaction, objects to be tracked are expected to occlude each other in the field of view of the camera observing them. A naive approach would be to employ a Set of Independent Trackers (SIT) and to assign one tracker to each object. This approach scales well with the number of objects but fails as occlusions become stronger due to their disjoint consideration. The solution representing the current state of the art employs a single Joint Tracker (JT) that accounts for all objects simultaneously. This directly resolves ambiguities due to occlusions but has a computational complexity that grows geometrically with the number of tracked objects. We propose a middle ground, namely an Ensemble of Collaborative Trackers (ECT), that combines best traits from both worlds to deliver a practical and accurate solution to the multi-object 3D tracking problem. We present quantitative and qualitative experiments with several synthetic and real world sequences of diverse complexity. Experiments demonstrate that ECT manages to track far more complex scenes than JT at a computational time that is only slightly larger than that of SIT.

1. Introduction

We are interested in tracking the full state of a scene consisting of multiple moving and interacting objects. We consider rigid or articulated objects which interact in front of a static RGBD camera. An example scenario is that of observing hands interacting with several objects in assembly/disassembly tasks. In such a context, the full state of the scene at a given time consists of the 3D position and orientation (3D pose) of all rigid objects as well as all the degrees of freedom (DOFs) of the articulated objects. Clearly, the accurate and robust recovery of this state is of paramount importance towards developing higher level interpretations, which is the ultimate challenge in scene understanding and the far sought goal of computer vision.

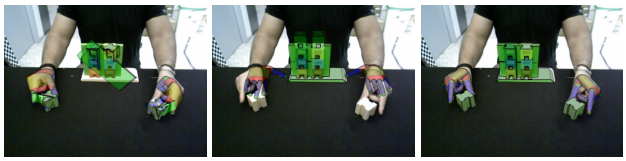


Figure 1: Tracking the full articulation of two hands and the 3D pose of 15 rigid objects in a toy disassembly task is a 159-D problem. The proposed ensemble of collaborative trackers (ECT, right) was successful in tracking this highly challenging scene, from RGBD input, while the other variants (SIT, left and JT, middle) failed right from the start.

A fundamental difficulty stems from the occlusions which occur when the physical 3D space is projected onto the 2D image of the camera observing the scene. If no such occlusions exist the whole problem can be easily decomposed. Assuming the availability of a tracker for each and every object, we may consider a Set of Independent Trackers (SIT) in a divide-and-conquer approach. This straightforward decomposition has an overall accuracy that depends on the accuracy of the individual trackers, alone. From a computational point of view, it scales well (linearly) with the number of objects to be tracked.

However, in the most interesting and most frequent case, objects do occlude each other in the field of view of the camera. This is particularly true in the object manipulation case we are interested in, where the interaction of hands with objects results in strong hand-object and object-object occlusions. Occlusions have a catastrophic effect on the SIT approach. This is because, due to occlusions, each individual tracker is fed either with missing or with ambiguous observations of the object it tracks. To handle this problem, state of the art approaches [10, 11] suggest a Joint Tracker (JT) which performs optimization over the joint 3D state of all objects to be tracked. Thus, occlusions are not treated as a distractor but rather as a source of information that has an active role in tracking the state of the scene. The JT approach has been shown to accurately track scenes of up to 54 DOFs, involving two strongly interacting hands [11].

But, as the complexity of the scene objects grows[1], optimization becomes much harder, and, as demonstrated here, the resources required to achieve a constant level of tracking accuracy increase geometrically with the number of the objects to be tracked.

In this paper, we propose a hypothesize-and-test method to track, accurately and robustly, the 3D state of complex scenes, in which strong interactions result in significant occlusions. Model-based tracking constitutes the basis of the proposed approach, which stands between the extremes of SIT and JT, trying to combine the best traits from both worlds. As in SIT, we employ a number of trackers, one for each object to be tracked. However, these trackers are not independent but they rather form an Ensemble of Collaborative Trackers (ECT). Being collaborative, the individual trackers solve the problem in a synergistic manner, bringing ECT closer to the spirit of the joint optimization performed in JT. Collaboration between trackers has the form of exchange of intermediate information. More specifically, each tracker is delegated with the task of tracking a single object, while regarding the rest of the objects as being statically defined. At certain stages of the processing, each tracker broadcasts its intermediate tracking results to all other trackers which update their current knowledge of the state of the other tracked objects. This way, ECT achieves two things simultaneously. Firstly, as in SIT, the full, joint optimization problem is decomposed in a number of smaller problems of lower dimensionality. Secondly, as in JT, occlusions among interacting objects are effectively taken into account.

Several experiments with a variety of scenes and complexities have been carried out to assess the proposed ECT approach, quantitatively and qualitatively. In all cases, the obtained results were compared to those of the SIT and JT approaches. In a representative experiment (Fig. 1), we considered two hands as they disassembled a toy consisting of 15 rigid objects. Tracking the articulated motion of the hands and the 3D pose of all objects corresponded to a problem of 159 dimensions. It is shown that while both SIT and JT failed (for different reasons) to track this scene, the proposed ECT method provided an accurate solution at a speed that was only $2\times$ slower than SIT and $50\times$ faster than JT. Similarly, for the rest of the experiments and for every optimization budget, ECT outperformed both SIT and JT in tracking accuracy, while it was only slightly more expensive than SIT and far cheaper than JT in computational time.

2. Relevant work

Multi-object tracking and occlusion handling are two strongly connected problems. The respective literature lists several approaches that tackle both problems simultaneously. The overwhelming majority of these works track 2D regions of interest (ROIs) and, as such, their full review is out of the scope of this work. The dominant approach is to treat occlusions as a distractor, *i.e*. perform special occlusion detection and rectification while or after tracking, as in [2]. Similar in spirit is the approach proposed in [12]. That work also reviews the state of the art and the relevant literature in robust occlusion handling while tracking multiple objects in 2D.

Such methods are, by construction, limited in estimating 2D information, with some exceptions that only go as far as considering depth qualitatively (*e.g*. depth layering). In the current work we focus on methods for the 3D tracking of multiple interacting objects. *i.e*. we focus on quantitative reasoning in all spatial dimensions. In a class of such methods, bottom-up evidence, provided by strong discriminative tools, is fused into coherent interpretations through higher-level generative modelling. Hamer *et al*. [6] proposed a robust reconstruction method that performs 3D tracking of a hand manipulating an object. This method was based on strong bottom-up evidence that was used to identify occlusion, so that their effect was disregarded during inference. Then, hand pose hypotheses were constructed by generative means. Romero *et al*. [13] exploited their ability to realistically synthesize the outlook of a hand-object interaction scenario to also track a hand manipulating an object in 3D. A non-parametric discriminative model was generated from a large synthetic dataset. This model was used to track hand pose from image sequences through classification. Inference over hand poses close in time were regularized so as to adhere to some smoothness criteria. While both methods set the basis for robust hand-object tracking, their extension to other types of interaction or their extension to tracking more objects is not straightforward.

Other methods go beyond treating occlusions as a distractor by explicitly accounting for them in interaction models. These methods are generative in nature and operate on raw or on slightly preprocessed input. Tracking is treated as an optimization problem that involves an objective function that quantifies the discrepancy between hypothesized interpretations of a scene and its actual observations. Oikonomidis *et al*. [10] tracked jointly an object and the hand manipulating it, in 3D. They considered an interaction model that directly accounted for potential occlusions and the fact that two different objects cannot occupy the same physical space. The same line of reasoning has been successfully applied to the more challenging problem of tracking two strongly interacting hands [11]. In both [10, 11] black box optimization was employed. Ballan *et al*. [3] followed the same general principle for tracking two hands in interaction with an object. However, they incorporated stronger preprocessing of their input that was based on elaborate dis-

[1] In this context, the complexity of a scene is quantified as the number of parameters/DOFs that represent its state.

criminative modelling and they considered the Levenberg-Marquardt algorithm for performing optimization. Interestingly, in all works it is acknowledged that inference over parts is more successful when all constituents are considered jointly, through their accounted interactions.

Multi-object tracking has also been approached as a Bayesian filtering problem. The Probability Hypothesis Density (PHD) filter is a prevalent approach to the multi-object tracking problem. PHD, and generalizations *e.g*. cardinalized PHD (CPHD) [16], have been used to tackle 2D and 3D trajectory estimation problems for point entities. These frameworks have not been used for problems with nature related to ours. We believe the reason for this, among others, is the non-triviality in handling highly articulated entities. This is exemplified in the filtering-based 3D hand tracking method in [15].

The works in [7, 8, 14] are most relevant to our proposal by providing 3D positions of multiple objects across time. Kim *et al*. [7] performed simultaneous camera and multi-object pose estimation in real time by exploiting SIFT features. However, the bottom up nature of the work allows for limited robustness and extensibility. Salzmann and Urtasun [14] derived a convex formulation over a physical model of interaction of multiple objects that enabled tracking in weak perspective projection scenarios. Despite the beneficial optimization traits, it is not straightforward to extend this approach while preserving convexity. In previous work [8], we were able to track multiple entities, a hand and several objects of known structure, from RGBD sequences by employing the physics-powered single actor hypothesis. This hypothesis distinguished entities into being active or passive, and allowed for virtually arbitrary counts of passive objects to be tracked without increasing the search space. However, the same did not hold for active entities, where the same issue as with [10, 11] applies. In contrast to the above mentioned state of the art approaches, the proposed Ensemble of Collaborative Trackers performs multi object 3D tracking that scales well with the number of active objects to be tracked.

3. Method

The proposed method estimates the 3D state of multiple interacting objects through tracking them across the frames of visual input obtained by an RGBD sensor. Tracking is model based, *i.e*. the appearance and the motion of the objects to be tracked is captured in a predefined forward model. The parameters of this model are connected to observations through an objective function that acts as a compatibility measure between hypothesized model instantiations and actual observations. Inference then amounts to identifying the most compatible model instantiation. This is achieved through black box optimization.

In notation, let a scene comprise N entities whose joint state is given by the vector $x \in X$. Let also $\mathbf{M}$ be a forward model which maps such states into a feature space F:

$$f = \mathbf{M}(x), f \in F, x \in X. \tag{1}$$

Given that there exists a process $\mathbf{P}$ which maps actual observations o to the same feature space F and a prior term $\mathbf{L}$ which describes how unlikely hypotheses are regardless of observations, we can formulate the following function $\mathbf{E}$

$$\mathbf{E}(x, o, \hbar) = \|\mathbf{M}(x, \hbar) - \mathbf{P}(o, \hbar)\| + \lambda \mathbf{L}(x, \hbar), \tag{2}$$

to quantify the discrepancy between actual observations o and a hypothesized scene state x, given the tracking history $\hbar$, *i.e*. observations and state estimations for all previous frames. Then the problem of estimating the state of the scene for the current frame reduces to finding the minimizer parameters s of $\mathbf{E}$ for given observations o:

$$s \triangleq \arg\min_{x} \mathbf{E}(x, o, \hbar). \tag{3}$$

We exploit temporal continuity in adjacent frames and do not perform the minimization globally. Instead, we minimize locally, in the vicinity of the solution estimated for the previous frame. Minimization is performed as in [11], by employing Particle Swarm Optimization (PSO). PSO searches for the optimum by evolving a population (particles) of initial hypotheses, over time (generations). As these hypotheses are evolved they are scored by invoking the objective function $\mathbf{E}$. The more invocations allowed the better the chance that PSO will converge to the true optimum of $\mathbf{E}$. However, computational cost increases in this direction.

In this work, and with respect to Eq. (2), we incorporate the data term (left term of Eq. (2)) of [9, 11], as a generic way to connect observations with hypotheses. *I.e*., we employ rendering to map hypotheses to depth maps, so that they become comparable to observations, and we differentiate in order to compute compatibility. We also make use of the prior term (right term of Eq. (2)) to penalize object collisions, as space cannot be shared among objects. We employ a penalty term which amounts to the total penetration depth for all pairwise collisions in a hypothesized 3D configuration of objects [5]. This should be contrasted to [11] which penalizes penetration between adjacent fingers alone.

3.1. Tracking with a Joint Tracker (JT)

Unless all entities are considered jointly, the corresponding forward model cannot capture their interaction and, therefore, cannot predict possible occlusions. The state of the art approaches [10, 11] tackle the problem of Eq. (2) directly. The configuration space of all objects combined is considered as the joint hypothesis space. This allows for the objective function of Eq. (2) to regard all entities simultaneously and to account for their interactions.

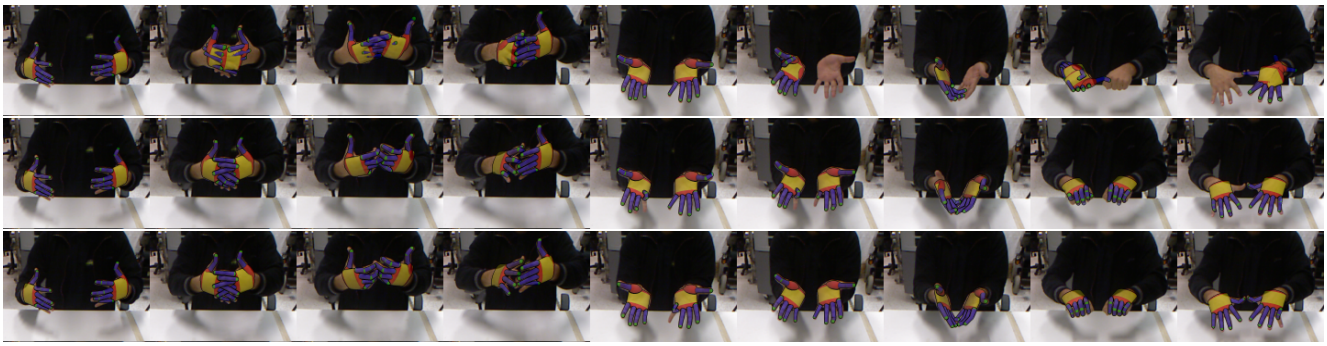


Figure 2: Tracking two interacting hands. The results of SIT (top row), JT (middle row) and ECT (bottom row), are compared for 50 particles and 50 generations per hand.

Two problems are associated with this approach. First, computing interactions and features for a large number of objects at every invocation of the objective function is computationally expensive. Second, as the search space grows, it becomes increasingly harder to practically solve the optimization problem. In fact, experimentation shows that JT does not scale well with the number of objects to be tracked and that it is thus incapable of handling the complexity of the scenes we are interested in.

3.2. Set of Independent Trackers (SIT)

An oversimplification of the problem would be to consider all objects disjointly. In detail, multiple independent problems are solved, where each one addresses the tracking of an individual object in isolation. Thus, the hypothesis space for each tracker is the configuration space of a single object. This yields a method that is significantly faster than JT due to the lossy decomposition of the big problem into multiple smaller ones. In case that there are no occlusions among the objects to be tracked, the quality of the solution (*i.e*. tracking accuracy) is determined by the accuracy of the solution of each individual problem. However, as interaction between objects and the resulting occlusions increases, the method is expected to deliver tracking results of progressively lower accuracy. This is because occlusions contaminate the observations of each individual object with missing or ambiguous evidence. Formally, the computed objective function, due to the lossy decomposition, no longer corresponds to Eq. (2), and therefore the respective minimizer does not respect joint constraints, such as the required mutual exclusiveness in the allocation of observations. This has been demonstrated experimentally in [11] where the JT approach managed to successfully track two hands in a dataset featuring two hands in strong interaction. In contrast, an independent single-hand tracker (SIT approach) performed accurately when there were no occlusions between hands, but, as soon as interaction started, it failed to recover the correct state.

3.3. Ensemble of Collaborative Trackers (ECT)

Our proposal is a middle ground between SIT and JT. As in SIT, we employ multiple trackers, each of which is responsible for tracking the state of a single object. However, the trackers are not independent. On the contrary, each of them considers the state for the rest of the objects to be static for the current frame, according to the most updated view of the other trackers. Reversely, each tracker broadcasts the estimated state for the associated object as soon as this becomes available, and this information is regarded as static by the rest of the trackers during their very next optimization step. Formally, this still allows for the full consideration of all possible interactions since the invocation of Eq. (2) regards the combined state. At the same time (a) optimization is clearly separated for each object, yielding a clear scalability profile and (b) less computations are required for the interactions and features, because the most part regards static information that needs only be computed once per frame.

The rationale behind this choice is that by breaking down the problem into multiple smaller ones optimization should be guaranteed, as in SIT, but at the same time, interactions will also be considered. In contrast to SIT, ECT has no ambiguity issues. The appropriate parts of observations that correspond to other trackers are already allocated to them, through the consideration of the broadcast results, leaving only the part of observations that correspond to each tracker, plus some noise that is introduced by the one-frame lag of these broadcast results. It is shown experimentally that the level of this noise is well inside the trackers' tolerance. Notably, ECT requires no more assumptions than JT does.

3.3.1 Decomposition of computations

In each frame, each tracker of the ensemble needs to estimate the state of the object it is associated with. To do so, it renders and evaluates object pose hypotheses (dynamic state) in a context that is formed by what has been estimated

for the rest of the objects, from the rest of the trackers, in the previous frame (static state). Essentially, dynamic state corresponds to the optimization task performed by each tracker and static state corresponds to a frozen state of the scene as it has been estimated so far. This distinction is quite important because it leads to considerable computational performance gains.

For each tracking frame the terms in Eq. (2) that are related to static state are precomputed. Then, during optimization for the same frame, the precomputed terms are fused with computations for the dynamic state, in order to properly assemble an invocation to the designed objective function. Given that dynamic state accounts typically for a small fraction of the scene (*i.e.*, one out of many objects), the computational gains are high.

Object rendering regards both static and dynamic state. At the beginning of each tracking frame, the static state of every collaborative tracker (*i.e.* the most up to date view of the scene according to the rest of the trackers) is rendered once into a depth map and stored. For each hypothesis generated during optimization (dynamic state), the corresponding depth map is fused with the stored one through z-buffering. The final depth map is identical to what the rendering of the entire state would have yielded. Nevertheless, it is much cheaper computationally during optimization.

Collision checking (as a prior term $\mathbf{L}$ in (2)) also regards both static and dynamic state. The total penetration depth TPD for a collection of shapes is defined as the sum of all pairwise penetration depths PD:

$$\begin{aligned}\mathrm{TPD}\left(h\right) &= \sum_{x,y\in h} \mathrm{PD}\left(x,y\right) = \\ &= \sum_{x,y\in h_s} \mathrm{PD}\left(x,y\right) + \sum_{x,y\in h_d} \mathrm{PD}\left(x,y\right) + \sum_{\substack{x\in h_s \\ y\in h_d}} \mathrm{PD}\left(x,y\right),\end{aligned} \tag{4}$$

where h is a 3D configuration hypothesis that regards multiple entities, h_s is the part of h that regards static state and h_d is the part of h that regards dynamic (per collaborative tracker) state, such that $h = h_s \cup h_d$ and $h_s \cap h_d = \emptyset$. The computations that regard h_s alone need only be computed once at the beginning of each tracking frame. During optimization this precomputed term is simply added[2] to the penetration depth computations for the rest of the pairs.

It should be noted that for both ECT and JT the objective function is invoked over the entire state of the scene. However, the same invocation to the same objective function is computationally cheaper for ECT, because for JT there is no static/fixed state whose processing could be re-used.

4. Experiments

All experiments were executed on a machine with a quad-core Intel i7 920 CPU, 6 GBs RAM and a 1581GFlops Nvidia GTX 580 GPU with 1.5 GBs RAM. For the image acquisition process we employed a Kinect sensor and the OpenNI framework. Acquisition was performed at a $30fps$ rate. For collision checking complex/concave objects were decomposed into convex parts. Convex decomposition was performed using the method in [5] as implemented in the CGAL library [1]. Collision checking was performed using the Bullet physics simulator [4]. During optimization, λ was set to 1 when $\mathbf{L}$ amounted to penalization of adjacent finger interpenetration (measured in radians), and 0.001 when $\mathbf{L}$ amounted to the total penetration depth penalty (measured in millimeters). Videos with representative results from all reported experiments are available at `http://youtu.be/SCOtBdhDMKg`.

4.1. Quantitative analysis

The tracking performance of JT, SIT and ECT was compared in a series of experiments where ground truth was available. The establishment of ground truth in real-world acquisition is known to be hard, especially for the case of hand tracking [9–11]. Therefore, we employed the common practice of generating synthetic datasets, where ground truth establishment was guaranteed. The fact that in that case observations were ideal is irrelevant, as the focus of this experiment lies in the quantitative comparative evaluation of the three tracking methodologies. Their efficacy in real-world scenarios pertaining to real noise, is demonstrated in Sec. 4.2. Efficacy was computed for various configurations of PSO. This configuration amounted to specifying the number of particles and generations. The product $budget = particles \times generations$ yielded the number of objective function invocations during the corresponding tracking frames.

Each of the employed datasets involved objects of the same type. Thus, the budget allocated to each tracker of SIT and ECT was identical. For experiments to be fair, at any given point of comparison the three tracking methods were provided the same budget. If N entities were involved, and for a selection of p particles and g generations, each tracker of SIT and ECT was allocated a budget of $p \times g$ objective function invocations. Thus, for these two variants each tracking frame amounted to $N \times p \times g$ objective function invocations. JT was allocated the same budget, by maintaining the generation count at g and by setting the particles count to $N \times p$.

Quantitative experiments were conducted across a 2D grid of budgets, that was generated by varying particle and generation counts. At every point of that grid several tracking experiments were conducted to balance effects stemming from the stochastic nature of PSO.

For each experiment we computed an accuracy measure E, that amounted to the average 3D registration error, in millimeters, between the true configurations of the entities

[2]The term remains constant during optimization and it can be left out.

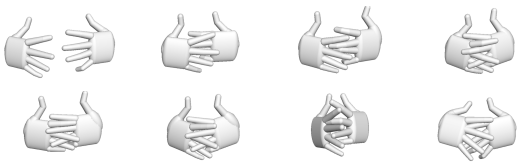


Figure 3: Frames of the two-hands tracking synthetic dataset used here and in [11] for quantitative analysis.

(a) radius of $120mm$ (b) radius of $250mm$

Figure 5: Frames from the rotating bottles synthetic dataset.

and the tracked configurations, for all frames and entities. Registration error of rigid entities amounted to the mean value of the registration error of each point of the corresponding 3D model. For the articulated entities the mean registration error across all joints was considered, as in [11].

4.1.1 Tracking two hands

The problem of tracking two interacting hands in 3D amounts to solving a 54-D (27 dimensions per hand) problem, for every tracking frame. This problem has already been tackled in [11] which in this context can be viewed as the JT method. This is compared against SIT and ECT on the synthetic dataset that was used in the quantitative analysis of [11] (Fig. 3). This dataset consisted of 300 frames showing two hands engaged in increasingly strong interaction. Thus, it was more probable that tracking accuracy deteriorated at later frames, as inaccuracies accumulate and drift increased. During optimization, we used the finger interpenetration penalty as a prior.

A budget grid was considered, where particle counts varied in the range $(0, 100]$ and generation counts varied in the range $(0, 50]$. The corresponding results are shown in Fig. 4. Apparently, ECT always outperformed both SIT and JT. Another striking result is that, on average, SIT was not much worse than the rest of the methods. The details in Fig. 4(c) yield an explanation. The accuracy for the JT method gradually degraded as the interaction complexity increased, due to accumulated drift. SIT and ECT had very similar behaviours, being more accurate on average than the JT method, until the two hands interacted (frame 100). After this point, the accuracy for the SIT deteriorated quickly, since it did not account for the intense interaction. The ECT method was unaffected by the increase in interaction complexity. We attribute the superiority of ECT against JT to (a) the explicit isomerization of optimization budget and (b) the fact that a difficult problem was broken down to two easier ones.

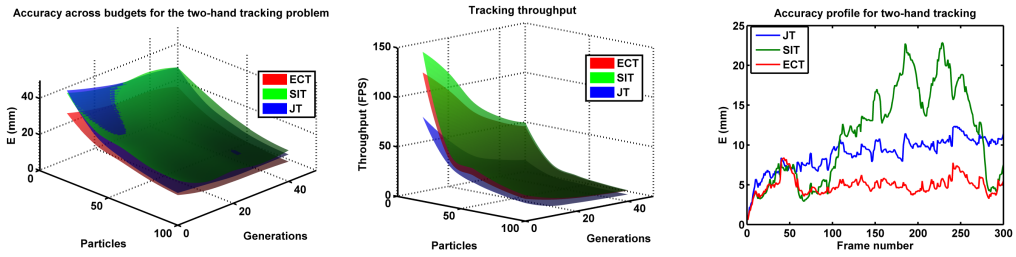
4.1.2 Multi-object tracking

The problem of tracking two interacting hands included the following challenges: (a) the problem dimensionality was high and (b) the articulated nature of the entities yielded a difficult problem to solve, due to interaction intensity. While (a) remains a constant pursuit, since we are targeting large problems, more insight can be provided by modulating (b). We therefore preserved the dimensionality of the problem and instead of considering two complex entities we regarded a larger number of simpler entities.

We animated 8 instances of a spraying bottle model as if they stood equidistantly on a rotating turntable (Fig. 5). By modulating the radius of the turntable, occlusions became lighter (larger radii) or heavier (smaller radii), *i.e*. occlusions occurred over smaller or bigger areas and included less or more objects simultaneously. The motion of the bottles was rigid, therefore for each bottle, during tracking, the pose was sought, which amounted to 7 parameters (3D position and quaternion-based orientation). Thus, the total problem dimensionality was $8 \times 7 = 56$. As objects were by construction penetrating each other we employed neither the total penetration depth penalty nor any other.

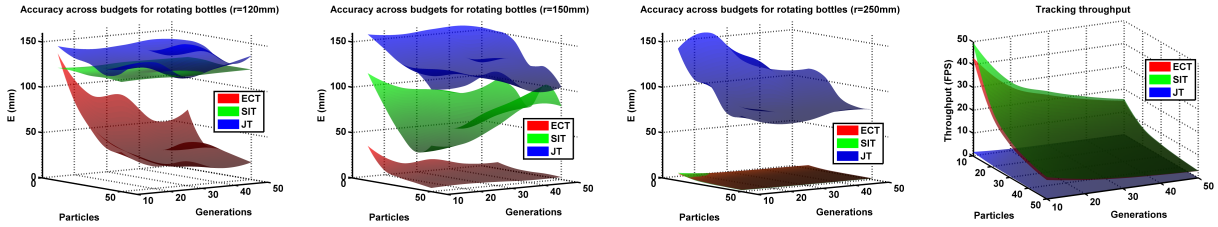
For this problem we considered a budget with particles varying in the range $(0, 60]$ and generations varying in the range $(0, 50]$. The results are shown in Fig. 6. As it can be verified, in all cases ECT yielded the most accurate performance. What is striking is that JT was always the least accurate. The accuracy of SIT varied between the accuracies of the two other methods, and degraded, as expected, as occlusions became more severe, *i.e*. in lower radii (Fig. 6(a)). For the case of the largest radius the performance of SIT and ECT was identical (Fig. 6(c)), which demonstrates a single tracker's adequate tolerance against contaminated observations. The order in tracking throughput was preserved, however, the differences were exaggerated due to the increase of the number of objects, rendering JT $25\times$ to $47\times$ slower, than ECT (Fig. 6(d)). The presented figures are greatly skewed in favor of ECT as the number of objects increases.

One thing to note is how much slower JT becomes as the number of objects increases. This is due to the ability of ECT to reuse computations over static state (see Sec.3.3.1). Considering large parts of static state dramatically decreases the amount of object pairs which require dynamic consideration, *i.e*. dynamic redefinition, rendering and comparison. The quadratic amount of object pairs (object to object), which require consideration in JT, becomes linear for ECT (object to scene).



(a) (b) (c)

Figure 4: Quantitative results for the two hands tracking regarding (a) accuracy, (b) tracking throughput and (c) average tracking performance across time.



(a) radius of $120mm$ (b) radius of $150mm$ (c) radius of $250mm$ (d)

Figure 6: Quantitative results for the rotating bottles dataset. (a)-(c) tracking error and (d) tracking throughput.

4.2. Qualitative analysis

To finalize the comparison of the three tracking methods we present tracking results that regard real sequences.

4.2.1 Tracking two hands

The dataset employed in the qualitative assessment of [11] was also used here, to compare the proposed ECT tracking methodology to that of JT and SIT. As in [11], we used skin color detection to identify the foreground. We used the total penetration depth penalty as a prior. The results are shown in Fig. 2. It is evident that SIT could not cope with the complexity of the problem. The performances of ECT and JT were comparable on average, with ECT yielding better results, overall. Besides the slightly better results, ECT was also up to $3\times$ faster than JT for larger budgets.

4.2.2 Disassembly tracking

We recorded a sequence of two hands disassembling a toy made of 15 parts, *i.e.* 1 base, 6 columns and 8 cubes (Fig. 1). Each hand corresponded to 27 parameters and each rigid part to 7, resulting in 159 total parameters. Because it was an assembly toy, parts fit in tightly, creating multiple and severe occlusions. Also, due to the similarity of the shape of the parts, one part could be mistaken for another, given that color information was intentionally left unexploited.

We employed SIT, JT and ECT to track this scene. The dataset consisted of more than 2900 frames. We identified the foreground by keeping only points non belonging to the surface of the table. We complemented with skin color detection to enhance poor foreground detection on the hands. We allocated a budget of 30 particles and 30 generations to each rigid part. For each of the hands we considered a budget of 50 particles and 50 generations. The budget was selected to be the minimum which would allow SIT to succeed if there were actually no interactions to account for.

SIT failed to track this complex scene, right from the start, as it did not account for interactions, which were dominant. JT failed too, as the budget was inadequate for exploring viable solutions in such a large space, despite the fact that most of the entities remained stationary for prolonged periods of time. In fact, during tracking, JT was unable to depart from the initialization hypothesis, which was maintained even across frames where data were clearly not supporting it. The results obtained from ECT are shown in Fig. 7. Evidently, ECT was successful in tracking the scene. Some tracking issues did occur, at moments where the poses of the hands generated ambiguous observations in the depth map. This, however, is an inherent problem to the approach of [9, 11] and does not relate to collaborative tracking (issues occurred when hands were fully visible). As far as execution times are concerned, SIT required $0.95s$ per frame ($1.05fps$), ECT required $2.06s$ per frame ($0.48fps$, $2\times$ slower than SIT) and JT required $106.7s$ per frame (less than $0.01fps$, $50\times$ slower than ECT). ECT is slightly more

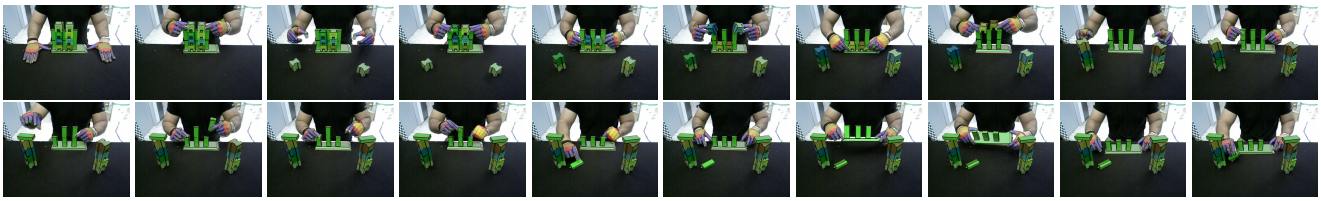


Figure 7: Representative results obtained from the proposed ECT approach on the toy disassembly dataset.

expensive in computational terms and for the exact same budget than SIT because of the additional consideration of the entire state at each individual tracker. ECT is far more inexpensive than JT, for the same budget, because of the decomposition and reuse of computations (Sec. 3.3.1).

5. Summary

We proposed a novel approach to the problem of tracking multiple active and interacting objects, in 3D, from RGBD input. Our proposal was to consider an Ensemble of Collaborative Trackers that run in parallel. Each of them tracks a single object, broadcasts its results to all others and exploits the results that are broadcast from the other trackers. The ECT approach was compared against the weak and computationally inexpensive (in relative terms) baseline method involving a Set of Independent Trackers (SIT) and the state of the art Joint Tracker (JT). Comparatively, the proposed method is almost as fast as SIT but far more accurate and faster than JT. Differences in accuracy and computational performance are strikingly widened in favour of ECT as the number of objects increases. Thus, ECT constitutes a practical and accurate tracking method for object counts and scene complexities that are far greater than what has been considered in the respective literature.

Acknowledgements

This work was partially supported by projects FP7-IP-288533 Robohow and FP7-ICT-2011-9 WEARHAP. The technical assistance of Avgousta Chatzidaki, member of FORTH/CVRL is gratefully acknowledged.

References

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012.

[3] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012.

[4] E. Coumans. Bullet game physics simulation, 2011.

[5] P. Hachenberger. Exact minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra in convex pieces. *Algorithmica*, 55(2), 2009.

[6] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *ICCV*, 2009.

[7] K. Kim, V. Lepetit, and W. Woo. Keyframe-based modeling and tracking of multiple 3d objects. In *ISMAR*, 2010.

[8] N. Kyriazis and A. Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *CVPR*, 2013.

[9] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*. BMVA, 2011.

[10] I. Oikonomidis, N. Kyriazis, and A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, 2011.

[11] I. Oikonomidis, N. Kyriazis, and A.A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012.

[12] V. Papadourakis and A. Argyros. Multiple objects tracking in the presence of long-term occlusions. *CVIU*, 114(7), 2010.

[13] J. Romero, H. Kjellström, C.H. Ek, and D. Kragic. Non-parametric hand pose estimation with object context. *Image and Vision Computing*, 2013.

[14] M. Salzmann and R. Urtasun. Physically-based motion models for 3d tracking: A convex formulation. In *ICCV*, 2011.

[15] E.B. Sudderth, M.I Mandel, W.T Freeman, and A.S Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPRW*. IEEE, 2004.

[16] B.N. Vo, B.T. Vo, N.T. Pham, and D. Suter. Joint detection and estimation of multiple objects from image observations. *IEEE Trans. on Signal Processing*, 58(10), 2010.