Boosting the Performance of Model-based 3D Tracking by Employing Low Level Motion Cues

Ammar Qammaz[1]
ammarkov@ics.forth.gr
Nikolaos Kyriazis[1]
kyriazis@ics.forth.gr
Antonis A. Argyros[21]
argyros@ics.forth.gr

[1] Insitute of Computer Science, FORTH,
N. Plastira 100, Vassilika Vouton,
GR70013, Heraklion, Crete, Greece
[2] Computer Science Department, University of Crete,
Heraklion, Crete, Greece

3D tracking of objects and hands in an object manipulation scenario is a very interesting computer vision problem with a wide variety of applications ranging from consumer electronics to robotics and medicine. Recent advances in this research topic allow for 3D tracking of complex scenarios involving bi-manual manipulation of several rigid objects using commodity hardware and with high accuracy. The problem with these approaches is that they treat tracking as a search problem whose dimensionality increases with the number of objects in the scene. This fact typically limits the number of the tracked objects and/or the processing framerate. In this paper we present a method that utilizes simple low level motion cues for dynamically assigning computational resources to parts of the scene where they are actually required. In a series of experiments, we show that this simple idea improves tracking performance dramatically at a cost of only a minor degradation of tracking accuracy.

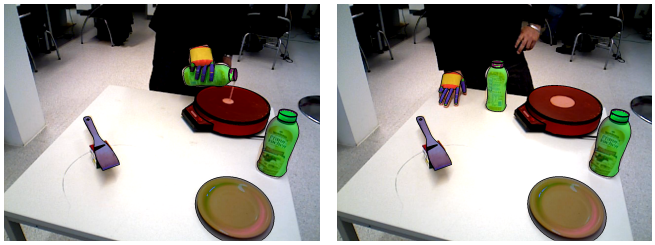
The works that are most related to ours are the approaches by Kyriazis and Argyros on top-down 3D tracking of multiple active objects from RGBD input [1, 2]. The methodological part of our contribution can be briefly described as an extra processing node in the pipeline of [2] which it extends. The tracking approach in [2], the Ensemble of Collaborative Trackers (ECT), regards a set of semi-independent trackers. Each tracker is associated with a distinct object in the scene. For an object to be tracked, a separate optimization problem is solved, one for each frame and each object. Each optimization problem is numerically solved using a black box optimizer, i.e. a variant of the Particle Swarm Optimization (PSO) algorithm. PSO treats the objective function as an oracle and queries it on purposefully evolved "guesses" in the search space, in order to find the optimum. The harder the problem is, the more guesses (budget, which is the product of PSO particles and PSO generations that need to be computed) are required for adequate accuracy to be achieved. For example, tracking the pose and the articulation of the hand amounts to solving a 27-parameter optimization problem. This is much harder than solving for the 3D pose of a rigid object (6 parameters). ECT [2] assigns a fixed amount of computational resources to each tracking sub-problem which depends on this notion of complexity and which is empirically estimated.

In this work, we quantify at run time, how hard the tracking of an object should be, not only based on its intrinsic complexity, but also based on its observed dynamics. An object that appears static in the recent temporal window requires less resources compared to an object whose state is more dynamic. Thus, change detection is performed on the image space of color intensities and depth measurements of the RGBD input. In more detail, for the next tracking frame, and the i^{th} tracker, a value $\mathfrak{m}_i$ is computed, which takes the value of 0 if the corresponding tracked object appears to be relatively static, and takes the value of 1 otherwise. For $\mathfrak{m}_i$ to be 1, any of the following need to be true:

- the mean value of the pixel-wise differences between the observations of object i and the back-projection of the last estimated configuration of object i is high enough,
- the kinetic energy of object i, as computed from so far tracked velocities, suggests a moving object,
- the amount of missing depth measurements changes substantially, from one frame to the next, which might me attributed to change in the slant of object i, or
- any of the above was satisfied in the recent past (damping).

Budget has the trivial minimum value of 0. Moreover, as it has been shown in [3, 4], a PSO budget of 64 particles and 64 generations suffices to track a hand, even in interaction with another hand. No more budget should be required to track simpler structures such as rigid objects.

The proposed dynamic budget allocation policy assigns a minimum budget B_{min} of 64 particles running for 4 generations to the objects that are static and a maximum budget B_{max} that never exceeds the aforementioned budget of 64 particles, 64 generations. thus, the budget $B_i, i = 1,\ldots,N$ for all N trackers is set to B_{max} if $m_i = 1$ or to B_{min}, otherwise.



(a) Pouring pancake mix (b) Waiting for the mix to cook

Figure 1: When making pancakes in real-life, several objects remain static for considerable time intervals. The proposed method can very efficiently skip through these idle times without sacrificing tracking accuracy.

Dataset	No of Objects	Object Complexity	Idle time	fps (ECT/F.ECT)	Error (ECT/F.ECT)
Two hands	17	Mixed	Medium	2.06 / 4.19	2.99 / 3.07
Spray	2	Mixed	Small	4.74 / 6.16	3.80 / 3.98
Cans	13	Low	Large	2.19 / 51.13	1.55 / 1.92
Pancake	4	Low	Large	5.62 / 86.22	2.30 / 3.10

Table 1: Experimental results and performance indicators.

In order to evaluate the proposed method we conducted several experiments on real (e.g. Fig. 1) and synthetic datasets. A summary of the results is presented in Table 1. In brief, several scenarios with different dynamics (number of objects, object complexity and idle time) have been considered. In all scenarios, the proposed method outperformed ECT [2] in tracking throughput, noticeably or even remarkably, while maintaining the accuracy levels (negligible error increase). Especially in scenarios with large idle times (e.g. cans, pancake, in Table 1) the increase in tracking throughput was dramatic ($15\times - 23\times$ speedup). A video showing qualitative experimental results is available at `https://youtu.be/nPru6PpWrK4`.

By focusing computational resources to active (i.e., non static) objects, tracking performance is not any more a function of the number of the involved entities but rather a function of the complexity of the actual motion and the interaction of objects present in the scene. Therefore, the proposed approach makes it possible to handle, accurately and at interactive framerates, scenes and scenarios that the current state of the art can only handle in offline mode.

Acknowledgements: This work was partially supported by projects FP7-IP-288533 Robohow and FP7-ICT-2011-9 WEARHAP.

[1] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 9–16. IEEE, 2013.

[2] Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3430–3437. IEEE, 2014.

[3] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, volume 1, page 3, 2011.

[4] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1862–1869. IEEE, 2012.