Optimal Assignment for Multi-Robot Tracking Using Motion Capture Systems

Nan Cai and Wolfgang Hönig

I. Introduction

Tracking for rigid bodies for robotic experiments is crucial, especially when working on advancing motion planners and controllers. The state-of-the-art are motion-capture systems from vendors such as VICON or OptiTrack, which can track rigid bodies with sub-millimeter accuracy at 250 Hz and beyond. These systems rely on infrared markers (passive or active IR emitters) that need to be placed in a a-priori known, fixed location on each rigid body. Each rigid body to track requires a unique marker configuration, limiting the usability of motion-capture systems to either few or very large robots. In previous work [1], custom rigid-body tracking has been introduced as follows: the motion-capture system outputs a pointcloud of the detected markers, the user has to provide an initial-guess of the pose of each robot, and a frame-by-frame iterative closest point (ICP) algorithm is used to find new poses for each rigid body independently. Subsequent (unpublished) improvements include a single-marker tracking mode, where each robot is assumed to carry only a single marker. Here, optimal linear assignment [2] is used at each frame, minimizing the total travel distance of all robots compared to the previous frame.

In this paper, we formalize the general problem where we want to track multiple robots (or rigid bodies), where some might be equipped with a single marker (and thus only their position may be estimated), and others use multiple markers (and thus the whole pose may be estimated). Surprisingly, the resulting combinatorial optimization problem is known to be NP-hard in general. We present two solution strategies that can solve these assignment problems efficiently in practice: an Integer Linear Program using a commercial solver, and a Conflict-Based Search-inspired algorithm and we empirically validate them. Our work has been already successfully used in experimental validation of aerial payload transport [3] and is publicly available as open-source (see footnote).

II. Approach

A. Problem Definition

We have N rigid bodies $\{r_i\}_{i=1}^N$, each of which has M_i markers rigidly attached to them, i.e., the relative transformation between the M_i markers and the i^{th} rigid body is known and fixed. At each time t, we are given the previous transformation of the rigid bodies in world frame $r_i.T \in SE(3)\ \forall i$, and a point cloud $\mathcal{P} \in \mathbb{R}^{3P}$. We want to estimate the current transformation of each rigid body $T_i' \in SE(3)\ \forall i$.

As is common in motion-capture systems, we do not assume any knowledge about dynamics, although a first-order filter might be used to reject outlier transformations. Note that the point cloud $\mathcal{P}$ is often noisy, i.e., might not contain all markers or contain spurious additional markers.

The prior existing solutions handle the case where $M_i \geq 4$ [1] or $M_i = 1$ (unpublished single-marker tracking). Here, we do not make such assumptions and allow mixed M_i.

Algorithm 1: Hybrid Update Function

Input: $\{r_i\}_{i=1}^N, \mathcal{P}$
Result: updated transformations $T_i'\ \forall r_i$
1 $S = \{\}$ ▷ *possible assignments robot r to group g with cost c*
2 **foreach** *robot* $r \in \{r_i\}_{i=1}^N$ **do**
3 **if** *r uses single marker* **then**
4 $\mathcal{P}_{near} = \texttt{KdtreeNearestSearch}(r.T, \mathcal{P})$
5 **for** $p \in \mathcal{P}_{near}$ **do**
6 $c = \|p - r.T\|; g = \{p\}; S = S \cup \{\langle r, g, c\rangle\}; T_{r,g} = p$
7 **if** *r uses multi marker* **then**
8 **for** $j = 0$ *to* L **do**
9 $T_{r,g}, group = \texttt{ICP}(r.T, \mathcal{P})$
10 $c = \|T_g - r.T\|; S = S \cup \{\langle r, g, c\rangle\}$
11 $A \leftarrow \texttt{optAssignmentWithGroupConstraints}(S)$
12 **for** $\{r, g\} \in A$ **do**
13 $r.T = T_{r,g}$

B. Multi-Robot Tracking

Our solution approach has two parts: first, we generate a number of potential assignment of the robots to groups of markers with an associated cost. Second, an optimal assignment problem is solved that minimizes the sum of the costs. The pseudo code is given in Algorithm 1. For the first part, the potential assignments for rigid bodies with a single marker are all points that are *near* the previously known location of each robot, where near can be the full point cloud or using a first-order filter (Lines 3 to 6). The cost is the L2 norm with respect to the previous estimate, the "group" is the single marker considered, and the solution, if assigned, is the position of that single marker. Similarly, for cases where the whole pose can be recovered, we run iterative closest point (ICP) L times to find potential marker groups g and the resulting transformation (Line 9).

The output of the first phase is a set S that contains tuples $\langle r, g, c\rangle$, where r is the rigid body to track, g is a set of markers that belong to the robot (a marker "group"), and c is the cost if the marker group is assigned to the robot. The resulting set is used to define an *optimal assignment problem with group constraints*, as formally introduced next.

All authors are with Technical University Berlin.

Code: standalone https://github.com/IMRCLab/librigidbodytracker, ROS 2 https://github.com/IMRCLab/motion_capture_tracking

The research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 448549715.

C. Optimal Assignment with Group Constraints

Algorithm 2: CBS-based solution

/ Main changes compared to CBS are highlighted. */*
Input: $\{r_i\}_{i=0}^{N}, \{g_i\}_{i=0}^{G}, \mathbf{C}$
Result: A
$\mathcal{S} \leftarrow$ `Node`$(solution : \emptyset, constraints : \emptyset)$ ▷ *Root node*
$\mathcal{S}.solution \leftarrow$`MFMC` $(\{r_i\}_{i=0}^{N}, \{g_i\}_{i=0}^{G}, \mathbf{C})$
$\mathcal{S}.cost \leftarrow$ `GetSolutionCost`$(\mathcal{S}.solution)$ ▷ *Update cost*
$\mathcal{O} \leftarrow \{\mathcal{S}\}$ ▷ *Initialize open priority queue*
while $|\mathcal{O}| > 0$ **do**
  $P \leftarrow$ `PriorityQueuePop`$(\mathcal{O})$ ▷ *max agents & Lowest cost*
  $t \leftarrow$ `FindConflict`$(P.solution)$ ▷ *multiple assigned tasks*
  if $t = \emptyset$ **then**
    return $P.solution$ ▷ *New solution found*
  else
    $\mathcal{C}_t = \{g | t \in \mathcal{G}_g \, \forall g \in \{1, \dots, G\}\}$
    foreach $g \in \mathcal{C}_t$ **do**
      $\mathcal{D} = \mathcal{C}_t \setminus \{g\}$ ▷ *Extract constraints*
      $P' \leftarrow$`Node` $(\emptyset, P.constraints \cup \mathcal{D})$
      $P'.solution \leftarrow$ `MFMC` $(\{r_i\}_{i=0}^{N}, \{g_i\}_{i=0}^{G} \setminus P'.constraints, \mathbf{C})$
      $P'.cost =$`GetSolutionCost` $(P'.solution)$
      `PriorityQueueInsert` $(\mathcal{O}, P')$

We consider a variant of the optimal linear task assignment problem, where each agent picks a set of tasks (rather than an individual task) and has to serve all of them. Formally, we have N agents and G task groups with a cost matrix $\mathbf{C} \in \mathbb{R}^{N \times G}$. Each group consists of a set of tasks, i.e., $\mathcal{G}_i = \{t_i^1, t_i^2, \dots, t_i^{|\mathcal{G}_i|}\}$. We want an assignment $N \mapsto \mathcal{G}_i$ (in form of a binary matrix $\mathbf{A}$) from agents to groups, such that: i) the number of assigned groups is maximized, ii) the overall cost is minimized, iii) each task is at most assigned once.

For i) and ii), we add N additional special groups with a high cost ($c_{ng} = c_\infty \forall g > G$), indicating "no assignment". Let the set of all tasks be $\mathcal{T} = \bigcup_{i=1}^{G} \mathcal{G}_i$. Then for a task $t \in \mathcal{T}$, we can compute the set of groups that contain this task as $\mathcal{C}_t = \{g | t \in \mathcal{G}_g \, \forall g \in \{1, \dots, G\}\}$. The solution can be found by solving the optimization:

$$\min \sum_{n=1}^{N} \sum_{g}^{G+N} c_{ng} a_{ng} \quad s.t. \quad a_{ng} \in \{0, 1\} \; \forall n \; \forall g, \tag{1}$$

$$\sum_{n=1}^{N} a_{ng} \leq 1 \; \forall g, \quad \sum_{g=1}^{G+N} a_{ng} = 1 \; \forall n, \tag{2}$$

$$\sum_{n=1}^{N} \sum_{g \in \mathcal{C}_t} a_{ng} \leq 1 \; \forall t \in \mathcal{T}. \tag{3}$$

Here, the first constraint ensures that each group is assigned to at most one agent; the second constraint ensures that each agent is assigned to at exactly one group (potentially the artificial "no assignment group"), and the last constraint ensures that each task is at most assigned once. This formulation is a rectangular assignment problem with conflicts, which is known to be NP-hard in the general case [4] and can be solved with Mixed-Integer Programs or Branch-and-Bound solvers [5].

This optimization can be solved using an ILP solver such as Gurobi [6] or a graph-search-based algorithm, inspired by Conflict-based Search (CBS) [7], an algorithm more commonly used for multi-agent path finding (MAPF). We use a two-level search, see Algorithm 2. On the low-level, we compute optimal assignments without constraints (3), which can be efficiently solved for example using a max-flow-min-cost (MFMC) formulation. On the high-level, we identify conflicts, i.e. any assignments that lead to double-assignments of a task. We then impose constraints to not include this assignment for each affected group assignment.

TABLE I
CBS VS ILP/GUROBI ON SYNTHETIC DATA.

L	T	N	Alg: Gurobi (ms)	Alg: CBS (ms)
3	3	5	0.533	**0.266**
3	3	10	**0.797**	0.825
3	3	15	**1.136**	2.577
3	4	5	0.486	**0.241**
3	4	10	1.082	**0.713**
3	4	15	**1.538**	6.200
3	5	5	0.930	**0.135**
3	5	10	**1.386**	1.474
3	5	15	**1.587**	6.054

TABLE II
PERFORMANCE ON REAL DATA.

N	M	Runtime		
		t (ms)	≥10 ms	CBS part
2	$2 \times 4 = 8$	0.29	0.14%	3.0%
3	$1 \times 4 + 1 \times 3 + 1 \times 1 = 8$	0.19	0.07%	4.7%
4	$4 \times 1 = 4$	0.02	0.00%	64.2%
4	$2 \times 4 + 2 \times 1 = 10$	0.30	0.12%	5.7%
4	$4 \times 4 = 16$	0.42	0.11%	3.5%
8	$4 \times 4 + 4 \times 1 = 20$	0.42	0.12%	6.4%
8	$4 \times 4 + 1 \times 3 + 3 \times 1 = 22$	0.56	0.26%	5.2%

III. EXPERIMENTS

To benchmark the efficiency of CBS-based assignment and Gurobi solver, we generate a synthetic dataset with varying numbers of agents, task groups, and tasks per group. L is the maximum number of groups under one agent r_i. T represents the maximum number of tasks in $\mathcal{G}_i$. N indicates the maximum number of agents in one test. As shown in Table I, the CBS-based assignment is faster when the scenario is simple, while Gurobi is faster in others.

In Table II, we conduct motion-capture tests with varying numbers of robots and marker configurations. The results show that CBS-based assignment achieves a median runtime below 1 ms across all tested configurations, with CBS contributing only a small fraction of the overall computational cost. Additionally, the fraction of frames exceeding 10 ms processing time remains negligible, ensuring real-time performance.

IV. CONCLUSION

We provide algorithms and software to track teams of robots with a motion-capture system. Unlike the vendor-specific software, our approach can use identical marker configurations on each robot, and supports cases where only the position, not the full pose, is required. Our approach is available as open-source standalone and as a ROS 2 package.

References

[1] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3299–3304.

[2] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Society for Industrial and Applied Mathematics (SIAM), 2009, 404 pp., ISBN: 978-0-89871-775-4.

[3] K. Wahba, J. Ortiz-Haro, M. Toussaint, and W. Hönig, "Kinodynamic motion planning for a team of multirotors transporting a cable-suspended payload in cluttered environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 12 750–12 757.

[4] A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger, "Paths, trees and matchings under disjunctive constraints," *Discrete Applied Mathematics*, Workshop on Graphs and Combinatorial Optimization, vol. 159, no. 16, pp. 1726–1735, 2011.

[5] T. Öncan, Z. Suvak, M. H. Akyüz, and I. K. Altinel, "Assignment problem with conflicts," *Computers & Operations Research*, vol. 111, pp. 214–229, Nov. 1, 2019, ISSN: 0305-0548.

[6] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2024.

[7] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.