Towards a Context Enhanced Framework for Multi Object Tracking in Human Robot Collaboration*

Sharath Chandra Akkaladevi, Matthias Plasch, Christian Eitzinger, Andreas Pichler and Bernhard Rinner

***Abstract*— In a goal-oriented Human Robot Collaborative (HRC) scenario, where the goal is to complete an assembly process, a robust object tracker might not necessarily fulfill its functional role due to the dynamic nature of HRC. Moreover, for an efficient HRC, the functional role of the object tacker should not only be limited to localizing and tracking objects for robotic manipulation. It should also help to determine the current state of the assembly process and verify if the chosen action has been successfully performed and thus to enable an uninterrupted completion of an HRC assembly process. We present a Context Enhanced Framework for Multi Object Tracking, that i) allows uninterrupted completion of an assembly process, ii) improves the overall functional accuracy of the object tracker from 49 percent to 96 percent, and iii) enables the object tracker to handle multiple instance of multiple objects in a HRC setting.**

I. INTRODUCTION

To enable a robotic system to understand the circumstances under which it operates, and react accordingly in a cooperative fashion with the human in a human robot collaborative (HRC) scenario, is a challenging task [1]. An example of such a cognitive robotic system is shown in Fig. 1, where cognition arises from the closely coupled integration between the reasoning, simulating, planning and adapting behavior of the robotic system.

Recognition and localization of objects of interest and tracking them is of vital importance in order to perceive and understand the current situation in HRC. To facilitate manipulation of objects by the robot, the object localization and tracking needs to be performed in 3D (3 DOF position + 3 DOF orientation). However, abrupt object motion, occlusions, clutter, complex object shapes and noisy sensor data make tracking difficult. If we consider multiple objects and real-time computational requirements in HRC, tracking objects in 3D becomes even more complicated.

In the scientific literature, there are approaches (e.g., [4]) that are capable of robustly tracking multiple objects in 3D and in near real-time. However, when applied to real-world HRC assembly processes, such approaches fall short in achieving high performance in terms of their functional role. The functional role in this aspect does not concern the overall accuracy of the tracker, but is concerned with "how good" the tracker performed in aiding the robotic system to complete the HRC assembly process (AP). In such APs, the **functional role of the tracker** is not just limited to facilitate manipulation of objects. It is also responsible for determining the current AP state and also to verify the success of a performed action (since these steps involve/require localization/tracking of relevant objects). The role of the object tracker in determining the current state of AP is concerned with tracking the objects of interest and providing the suitable object 3D pose for manipulation. In case of verifying the action, the object tracker can also help in verifying the action consequence (e.g. to confirm that an object has been displaced as expected during the manipulation). This adds another layer of redundancy for the HRC assembly to confirm that the action has succeeded (or failed) and the assembly process can proceed accordingly.

If the tracking approach fails to correctly localize/track the object(s) of interest during a particular step of the AP, the manipulation fails and results in an incomplete AP. Therefore, in spite of having high accuracy, a tracking system might fail to achieve its functional role in the AP. One idea could be to solve such problems is to configure the tracker to focus on specific objects depending on the current step of the assembly process, thereby improving the functional performance (how well the functional role was achieved). However, extracting such information (for e.g., which object(s) to focus) is not trivial (and also might not be sufficient) due to the dynamic nature of HRC assembly processes. Additionally, in APs involving multiple instances of similar objects, it is even more complicated in extracting the required information. It is also very challenging for the

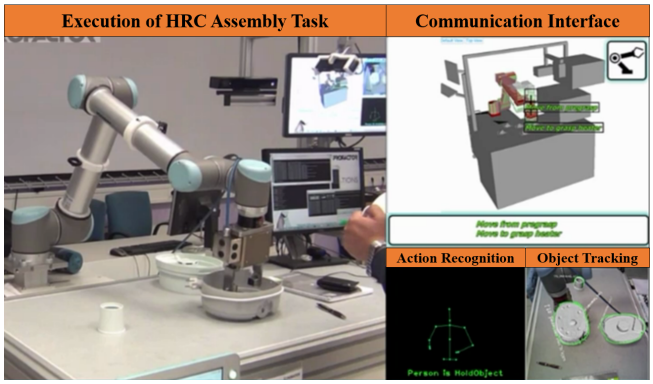

Fig. 1. A Robot manipulates objects of interest in coordination with a human operator in an integrated cognitive architecture, where the robot perceives, reasons, plans, executes and adapts. The image also depicts the object tracking, action recognition and communication interface modules that are integrated within the cognitive architecture for human robot collaboration.

S.C. Akkaladevi, M. Plasch, C. Eitzinger and A. Pichler are with Profactor GmbH, Im Stadtgut A2, Steyr-Gleink, 4407 Austria. email: sharath.akkaladevi@profactor.at

B. Rinner and S.C. Akkaladevi are with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Austria.

***This is a draft. For a complete version visit IEEE.org**

tracker itself to localize/track multiple instances of multiple objects.

The contributions of this paper can be summarized as follows:

1) development of a framework capable of extracting the relevant *context* in a dynamic human robot collaborative assembly process
2) improving the functional performance of object tracking by extending our previous work [4] to verify the current AP state and to verify the success of action execution
3) improving handling of multiple instances of identical objects in the AP

Note: In this work, the relevant information required to improve the functional performance of a module is termed as *context*.

The framework presented in this paper extracts the relevant *context* and can be applied to all the required modules (perception, planning, reasoning) of the HRC assembly accordingly. However to limit the focus, more emphasis is placed on the aspect of extracting *context* and its application to improve the functional performance of object tracking in the AP. The improved tracker is termed as **C**ontext **E**nhanced **M**ulti **O**bject **T**racker (CEMOT). The work also enables CEMOT to handle multiple instance of multiple objects in the AP.

The remaining part of the paper is organized as follows: Section 2 discusses the state of the art approaches on 3D tracking using RGBD data and approaches that use *context* driven object tracking are discussed. Section 3 presents the cognitive architecture for the HRC assembly process and the framework to extract the relevant *context* and Section 4 describes CEMOT. Section 5 summarizes the experimental evaluation. Section 6 concludes the paper with a discussion about future research

II. STATE OF THE ART

There are are only few approaches known in literature that deal with multiple object tracking using RGBD data and consider *context* information in an HRC assembly process. Therefore, the approaches that deal with 3D model-based object tracking using RGBD data are discussed first. Then multi target tracking approaches that consider *context* information are briefly described. Finally, few approaches that consider *context* enhanced object tracking in HRC assembly process are discussed.

Recent work in cognitive psychology and computer vision has shown that a statistical summary of the objects present in the scene can serve as an extremely effective source of information for contextual inference [5]. A recent survey on *context*-based information fusion is presented in [14]. One of the key challenges of tracking is to effectively verify if the object being followed by the tracker is really the required target object. Using *context* information is also an attractive strategy for object recognition [17], single target tracking [2][3], image captioning [26], multiple target tracking [15][16] and recognizing human activities [11][13]. The concept of mining auxiliary objects or local visual information surrounding the target to assist tracking is used in [12]. Maggio et al., [15] propose to exploit information about context-dependent events, such as objects entering the scene or reappearing after occlusion and spatially persistent clutter. However, these approaches use 2D image data as input and hence cannot be directly integrated into HRC that require object manipulation.

Ognibene et al., [18] integrate temporal and spatial contextual information to help predict and track human effectors with an active camera in a humanoid robot interaction scenario. The work in [19] shows how learning *context* to support tracking improves robot performance in various tasks. Most HRC approaches with task driven goals only consider object localization for manipulation and thereby do not consider the problems associated with dynamic environments, tracking and manipulation of real world objects [7][8][9].

The SMOT system [4] uses local *context* (for example, assumptions that an object cannot move more than a threshold between consecutive frames, direction of movement of the object) as additional information to verify the tracking results. The SMOT system as a standalone module is able to efficiently deal with partial occlusions, clutter and abrupt object motion. However, this approach faces difficulties when included into an HRC assembly process. During which the manipulated objects could be completely occluded, some objects disappear and reappear in the scene resulting in ghost detections and false positives. As mentioned in [6], the performance of multi object tracking is based on both goal-driven and stimulus-driven attentive and perceptual processes as well as spatial memory representation. Along these lines we extend the SMOT system by closely integrating it in an HRC cognitive architecture (using the framework to extract *context*) and transform it into the Context Enhanced Multi Object Tracking (CEMOT) system.

Unlike other HRC approaches [7][8][9], our system is capable of handling 3D tracking of multiple real-world objects (with multiple instances of each object type) in a goal driven assembly process. The required *context* for CEMOT system is provided through the tightly integrated cognitive architecture. This helps to improve the functional performance and as a result improves the tracking performance of the CEMOT system.

III. FRAMEWORK TO EXTRACT CONTEXT

A. The Cognitive Architecture

To deal with the HRC assembly process the robotic system should be enabled with cognitive capabilities. The architecture as shown in Fig. 2, consists of the following components a) **The Perception System** provides the real-time 3D tracking of objects with the help of the **Multi Object Tracker** [4][10] and current action performed by the human operator [23]. b) **The Planning and Execution System** generates plans for future actions to achieve the given task (goals). This includes task planning and scheduling for an efficient human robot collaboration. These generated plans

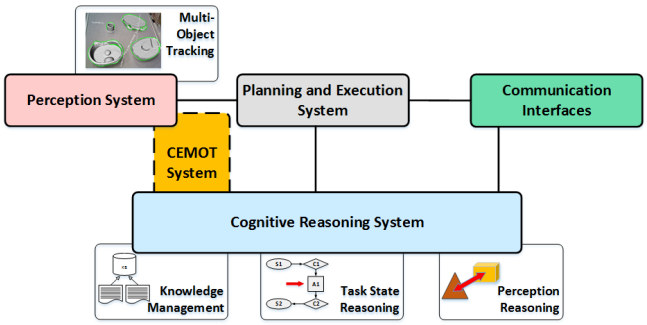

Fig. 2. Cognitive Architecture for HRC

need to be carried out in real world where the robot plans its path (path/navigation planning) and manipulates the environment accordingly. c) **The Communication Interfaces** provide a GUI-interface for human robot communication and simulations of the robot's planned behavior. d) **Knowledge Management (KM)** is based and extended upon KnowRob [21] to represent and abstract knowledge. e) **Perception Reasoning (PR)** understands and interprets the current state of the environment and the assembly task. f) **Task State Reasoning (TSR)** reasons about the current state of the AP by combining information about the current state (given by perception reasoning) and the assembly process knowledge (knowledge management), to make decisions to plan the next actions accordingly in coordination with the planning and execution system, to behave intelligently, to interact naturally with humans and aid in completing the task. The knowledge management, perception reasoning and the task state reasoning modules constitute the **Cognitive Reasoning System** in the architecture. The **CEMOT** system can be seen as a bridge between the object tracker (perception system) and the cognitive reasoning system and is explained in detail in Section IV.

B. Modeling Knowledge in HRC Assembly Process

An assembly process *(AP)* in its simplest form can be defined as a sequence of States S, a set of Events V and a set of Relations R. The set of States S defines the individual steps of the assembly process. The set of Events V drives the progress of the assembly process from one step to another. The Relations R specify the effect of a given Event V on a given State S in progressing the assembly process. A detailed formal description of the AP and its constituents is given in [20]. The architecture consists of **KM** that defines the corresponding data structures to manage and abstract the *knowledge of the assembly process*. This includes task state descriptions, robotic system configuration, capabilities of the robotic system and human operator, involved objects, their configurations and affordances, the properties of agent's (human, robot) actions and their corresponding effects on objects (see Fig. 3). The KM framework is an extended implementation of KnowRob [21] as KnowRob provides the following knowledge processing features: a) mechanisms and tools for action centric representation, b) automated acquisition of grounded concepts through observation and experience, c) reasoning about and managing uncertainty, and fast inference.

The knowledge is represented using ontologies (description logics) based on the Web Ontology Language (OWL). SWI Prolog is used for loading, accessing and querying the ontologies. The representation consists of two levels: **Classes** that abstract terminological knowledge (type of objects, events and actions) and **Instances** which represent the actual physical objects or the actions that are actually performed. **Properties** establish relations (links) between Classes, and these links are also valid for the Instances of the respective Classes. For example, Properties define if an $Agent \in \{Human, Robot\}$ can perform a particular action (defined in Classes) on/with a $Target \in \{Objects, Robot, Human\}$ [20][24].

The KnowRob framework provides a suitable basis (base ontologies) for modelling actions, objects of interest, and capabilities of humans and robots. A collection of Prolog rules are also provided for parsing ontologies and loading them into the Prolog database, thus making the ontology data accessible for database queries. We extended the base ontologies to express an HRC Assembly Process description. Moreover, Prolog rules were also extended to provide functionalities such as a) posting a snapshot created by the Perception System into the database, b) checking if recorded perception data fulfills the constraints of an assembly process state, c) projecting the expected outcome of an action that is planned for execution, and d) deriving the expected succeeding assembly process state. All these functions rely on Prolog queries (e.g. unification and proof search in the database, difference-list operations [22]) and ontologies (e.g. deducing facts which are not explicitely asserted in a database through so-called 'computables' [21] and ontology-reasoning).

C. Online Reasoning in the AP

An AP in HRC involves presence and manipulation of several objects. The AP consists of different steps, where each step requires a particular kind of manipulation on specific objects. For the robotic system to successfully complete the AP, it should a) determine the current state in the AP, b) choose/plan a necessary action to progress the AP, c) execute the planned action, d) verify if the action was successful; All these steps are iteratively executed until the AP is successfully completed (see Algorithm 1).

Determine the current state in the AP Given the assembly process and assuming it begins with the initial state, the TSR queries the knowledge management system for information regarding the current state (initial state) in the AP (see Algorithm 1). This data includes process state constraints that describe a) spatial relations between objects, present in the workspace, b) required states of the human and robot and c) Event descriptions that lead to subsequent assembly process states. Based on the given spatial relation constraints the TSR deduces the objects of interest for the given state. A snapshot of the current scene (object locations - Tracking, human and robot states - Action Recognition

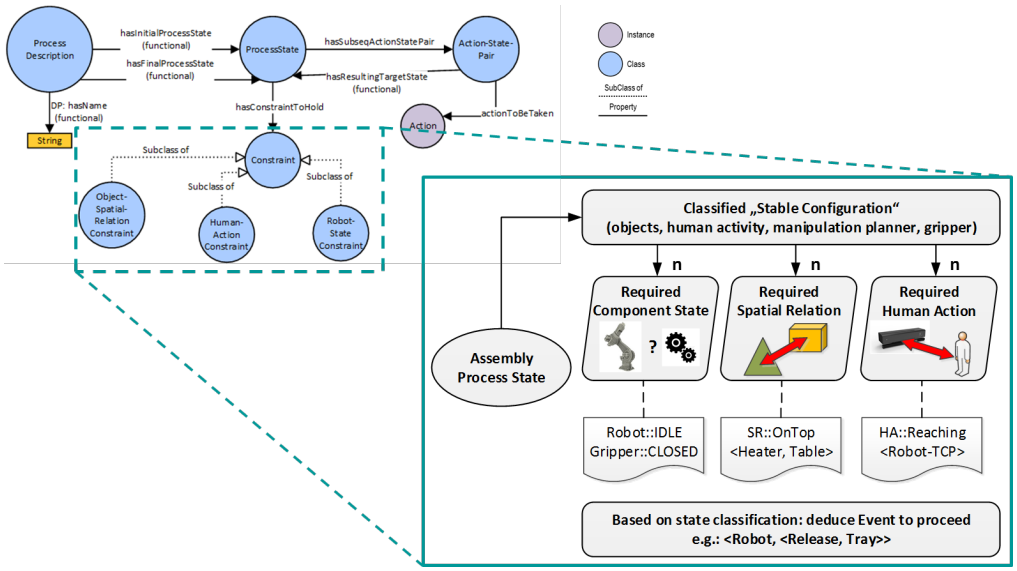

Fig. 3. Characterization of an individual state of the AP and its constituent parts (and the fashion in which knowledge is modeled for each state) that includes object spatial configurations, robot/human states w.r.t the AP environment

Set *processFinished* = false;
Set *executionError* = false;
Set *currentState* = *Initial State*;
while
processFinished = *false*&*executionError* = *false* **do**
  Query process state information from KM;
  Configure PR with given *context* information;
  Generate scene snapshot and post to KM;
  Trigger state verification;
  if *!Verification successful* **then**
    executionError = *true*;
    Continue;
  end if
  Reason *current state*;
  if *currentState* = *Final State* **then**
    processFinished = *true*;
    Continue;
  end if
  Decide action to progress towards goal;
  Parameterize action instance;
  Project action outcome;
  Make succ. state hypothesis(NextState);
  Configure PR with given *context* information;
  Trigger action execution;
  Perform two-step action verification (PES response and Action effects);
  if *!Verification successful* **then**
    executionError = *true*;
    Continue;
  end if
  currentState = NextState;
end while

Algorithm 1: Step-wise execution of AP including State Verification, Action Execution and Action Verification

and Robot Proprioception) needs to be created. The TSR triggers the PR with the given *context* information a) objects of interest available in workspace and b) robot and human in given state (e.g. IDLE state), and requests the validation of these constraints. The PR then waits for a stable response of all perceptions sources, i.e. Object Tracker, Action Recognition and Robot proprioceptive feedback to validate the *context* information provided by TSR. Afterwards, PR posts this snapshot into the Prolog database using a specific rule. Now the verification rule is triggered, which uses the KnowRob built in computable *comp_spatial* to verify if the spatial relation constraints are fulfilled and also compares detected human and robot states with the given process state constraints. The built in computable are functions that help verify the spatial relations of objects, given the current configuration of objects in the AP. If the verification succeeds, the given process state is assumed to be verified and the related Event descriptions are evaluated to deduce the next action to be executed.

The given Cognitive Architecture needs to deal with **multiple instances of the same object type** in the assembly process. In order to express a spatial relation constraint to be valid for a number of instances, we combined the expressiveness of OWL-Classes and their related OWL-Instances: We model a certain spatial relation of a certain pair of object types as an OWL-Class (e.g. 'Sphere-onTopOf-WorkTable'). To express a number of distinct configurations of this spatial relation, an instance of the considered OWL-Class is created and asserted with a OWL-Data-Property (e.g. an integer value) that describes the required quantity.

Planning of actions The physical execution of an action, deduced by the TSR, requires its proper parametrization based on the given assembly process knowledge. Each type of action is modeled as an OWL-Class in the assembly process specific ontology. An action type OWL-Class describes

the principal primitive (e.g. *PickAndHold* or *Insert*) and also describes the object types and targets affected by that action, as well as the type of actor that is capable of executing it. The related parametrization problem is described as finding concrete instances of objects, targets and the actor and generating a specific action instance. Considering the fact of multiple object instances available, e.g. for picking an object 'Sphere', the main question is the following: Which object instance shall be chosen? Our solution to this problem is to let the Planning and Execution System (PES) decide on selecting an appropriate instance. Even before triggering the execution, the TSR computes the expected action outcome, by projecting the potential action result with respect to the current state in order to acquire information on the expected changes in the environment (added / removed number of object instances, changed states of human and robot).

Execution of planned action For triggering the execution, TSR executes a Prolog rule to get all possible candidates for object instances, target instances, and actor instances available. This data is forwarded to the PES that triggers human action execution (notification on GUI) or robot execution, dependent on the actor type, and configures the PR accordingly to initiate verification of the action.

The Verification of action executed is performed in two steps. Firstly, the response of the PES (i.e. success or failure) is considered and second, the TSR tries to verify the whether the expected changes in environment did happen accordingly (e.g. object instances removed/added). This is performed by configuring the PR to check whether specific objects were removed / added at certain locations, or a human / robot state change has happened. If the action results could be verified, the TSR tries to compare the predicted state (using the given assembly process knowledge) to the perceived current state. From this step on, the procedure is repeated until a final state is reached.

IV. THE CEMOT SYSTEM

As mentioned in the introduction, the functional role of the object tracker is not just to facilitate object manipulation but also to aid in determining the current state of the AP and in the verification of the action executed. Even though object trackers like SMOT [4], track multiple object with high accuracy they cannot fulfill the functional role of determining the current state and verifying the executed action (evaluated in Section V). The reason is that SMOT system is statically configured, meaning that it can track a fixed set of multiple objects. However, dynamic changes like sudden appearance/disappearance of objects are often in HRC AP manipulation processes. Due to the presence of the human operator in the loop, the dynamic nature of the HRC AP is hard to predict.

To solve these problems, the CEMOT system (that acts as bridge between object tracker and cognitive reasoning system) shown in Fig.2 is used. CEMOT, first exploits the framework explained in Section III, to extract the relevant *context*. The *context* required by the CEMOT system includes, number of object types in the assembly process $numObjTyp$, where each object type is already known by means of a 3D model to facilitate localization and tracking. It also requires the number of object instances for each object type $numInstEachType$, this helps the tracker to deal with multiple instances of the same object. And finally it requires the type of operation (mode) to be performed on each object type instance $modeOfEachInst$. The different modes of operation of CEMOT are *Track*, *TrackAtpose*, *LocalizeROI* and *donotTrack* respectively.

A. Functional role of Tracker in AP

State verification in the assembly process To verify the current state of the assembly process, the TSR among other verification requires to confirm presence of a certain set of objects. The TSR with the help of PR gathers the required *context* (the object type, number of each object type, known location if any). If the known location for an object is given, CEMOT uses *LocalizeROI* to localize and track the object at that location. Otherwise, it uses the *Track* mode to localize the objects globally and then to track them globally [4]. When the tracker looses an object due to noisy sensor data, the TSR realizes this situation and provides the previous known pose of the object (only if the object was known to be static in the workspace). The CEMOT system then uses the *TrackAtpose* mode to track the object.

Action verification in the assembly process TSR needs to confirm the success of each individual action before proceeding to the next. Only relying on the proprioception of the robot might not be sufficient to robustly verify the success of the action. Hence, the TSR verifies the hypothesis of action projection with the help of CEMOT system. For example, if the hypothesis entails disappearance of a particular object, the *context* is prepared accordingly and communicated to CEMOT. The CEMOT system uses *LocalizeROI* mode, which attempts to localize the object within the ROI (previous known location of the disappeared object) provided. However, in this case the object will not be localized or would be localized with very low confidence and can be easily pruned. The results are then communicated back to the TSR which verifies the success/failure of the hypothesis. If the object was known to be moved to a new location, the CEMOT system is contacted to localize the object at the projected new location.

The CEMOT system only deals with objects relevant to the functional role (determining the state , verifying the action). It forgets all the other objects by using the *donotTrack* mode. This helps in reducing the computational time for the trackers as they can then only track/localize the objects of interest.

V. EXPERIMENTAL SETUP AND EVALUATION

The experimental setup depicted in Fig. 4a consists of a UR-10 robotic manipulator with 6 degrees of freedom, which is equipped with a SCHUNK electric parallel gripper. Two RGB-D sensors, Kinect 2 and Asus Xtion Pro provide depth data to the perception system, to enable human action recognition as well as object localization and tracking.

The evaluation of this work is done as follows:

TABLE I
ASSEMBLY PROCESS DESCRIPTION TO CLEAR WORKSPACE.
NOTATIONS: BASE(B), HEATER(H), TRAY(T), RING(R)

State S_i	Description	Event/Actions possible
S_0	$< B,H,T,R >$ on table	Human pick $< B >$ and place it into the Box
S_1	$< H,T,R >$ on table; $< B >$ not present	Robot pick $< H >$ and place it into the Box
S_2	$< T,R >$ on table; $< B,H >$ not present	Robot pick $< T >$ and place it into the Box
S_3	$< R >$ on table; $< B,H,T >$ not present	Robot pick $< R >$ and place it into the Box
S_4	$< B,H,T,R >$ not present	No action possible/required

1) Demonstrate the ability of SMOT and CEMOT in carrying out their functional role (determination of the current state of the assembly process and verifying the current action executed) in the assembly process
2) Showcase the ability of the CEMOT system in handling multiple instance of identical objects in the assembly process
3) Provide a qualitative comparison of the proposed CEMOT system against the state of the art that deals with human robot collaborative assembly processes

The comparison is carried out with two version of the cognitive architecture presented in Section III: one including the CEMOT system and one without it which is referred to as SMOT. All other aspects of the architecture remain the same. Due to the static integration between SMOT and the reasoning system, SMOT can only work in *Track* mode all the time. The dynamic integration in case of CEMOT allows it to work in the various modes as explained earlier.

As described in Table I, in the initial state S_0 of the AP, all objects are placed on top of the table. The TSR system given the AP knowledge has to verify if S_0 is really the current task state in order to proceed. TSR queries PR to verify S_0. PR then verifies the same with the help of the CEMOT/SMOT system. Given the type of objects (in this case B, H, T, R) and number of instances of each type, the *mode* of tracking is set to *Track*. The CEMOT system uses this configuration message and replies back with the tracked objects. If the number and type of tracked objects match the expectation of PR and as a result that of TSR, the assembly process is continued.

A total of 20 experiments were conducted, with 10 execution trials for each the SMOT and the CEMOT system. For both trial series the objects considered in the evaluation use case, were arranged similarly in order to provide similar conditions for both approaches. The AP is described Table I.

In order to provide a measure of functional performance, a collection of characteristic values, in this case determining the current state (**DCS**) of the AP and verification of the executed action (**VA**) were chosen and inquired during the experiments as shown in Table II and Table III for the SMOT and CEMOT system respectively. Additionally, the overall accuracy of the SMOT and CEMOT for each step of the AP is also measured.

In case of **DCS**, the role of object tracker is to correctly determine if a set of objects are really localized and tracked. Hence the statistical measure of a binary classification test *Sensitivity* is used to determine DCS. To determine **VA**, the role of object tracker is to correctly determine the action consequence, in this case (see Table I) to verify if action projection of disappearance of an object is really as such. Therefore, *Specificity* measure is used. In case of initial state, VA is not valid as the initial state is already assumed to be start of the AP and no action verification is necessary. The overall functional performance is the average of DCS and VA and is denoted as **OFR**. In case of the initial state, OFR measure is not valid.

A result reported by the object tracker is considered *valid*, if the detection result of the tracked object is stable ($relativechange \leq 10mm$) over a period of 10 frames. Since the object tracker is capable of tracking objects at $1.8ms$ [4], it can still deal with object movements that occur in an HRC assembly process.

For the evaluation of the SMOT and CEMOT system, only the *valid* results are considered. In the evaluation, true positives (TP) refer to reported detections at positions, which reflect the real situation (ground truth). A false positive (FP) is a reported detection at a position in the workspace where no corresponding object instance is located. False negatives (FN) refer to existing objects in the real world, but the system was not able to locate them. Finally, true negatives (TN) refer to object instances, which are not present in the work space and their absence is correctly confirmed. During the experiments, in the case the system (SMOT or CEMOT) is not able to verify the current state (DCS), the assembly process execution is canceled and the trial is deemed unsuccessful.

Both approaches (see Table II and Table III) show similar performance in the initial state S_0 (all objects are present). Since SMOT and CEMOT are configured to the same number of objects, the assembly state is correctly verified by both approaches. In some cases, both SMOT and CEMOT fail to verify the current state when in the initial state S_0 due to bad tracking results. This happens thrice for SMOT and twice for CEMOT. Hence that experiment had to be terminated, resulting in reduced trials for S_1. However, the performance of the SMOT system deteriorates and the performance of the CEMOT system improves as the assembly process progresses.

Though the SMOT system is able to verify the current state in some instances, it fails to verify the action executed as it is not capable to account for the true negatives (TN) as shown in Figure 4b-d. As the SMOT system is statically configured to track a fixed number of objects and cannot verify actions, the overall functional performance drops to an average value of 0.49. On the other hand, the CEMOT system is able to verify both the current state and the action executed as shown in Figure 4e-h. The overall functional performance of CEMOT amounts to 0.96. This implies that the SMOT system manages only $49 percent$ of the functional role, while CEMOT fulfills 96 percent of it. Also, the overall

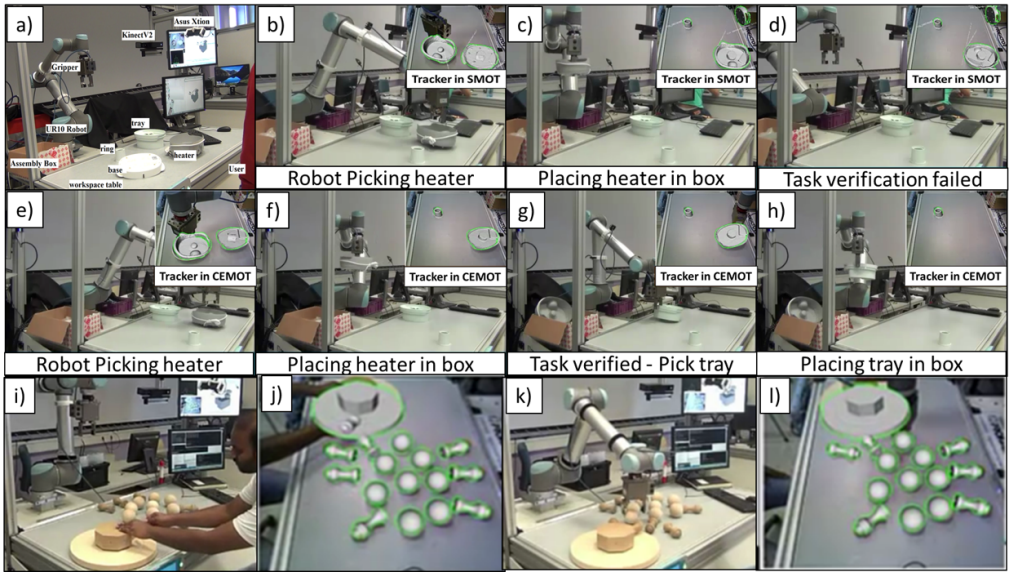

Fig. 4. a) Experimental setup including robotic manipulator, 3D sensor system, Human Machine interactive visualization, objects and human agent; b) to d) performance of the SMOT system in the assembly process - the SMOT system cannot confirm the executed action due to false positive and hence quits the assembyl process; e) to h) CEMOT system successfully verifies the current executed action and proceeds with the next steps in the assembly process; i) to l) Assembly process with multiple instances of multiple objects.

TABLE II

EVALUATION TABLE FOR THE SMOT SYSTEM

Assembly Steps	**SMOT**				
	N*OI=T	**DCS**	**VA**	**OFR**	**Acc.**
$S_{0\rightarrow 4objects}$	10*4=40	0.925	NA	NA	0.86
$S_{1\rightarrow 3objects}$	7*3=21	0.95	0	0.475	0.71
$S_{2\rightarrow 2objects}$	6*2=12	1	0	0.5	0.5
$S_{3\rightarrow 1object}$	6*1=6	1	0	0.5	0.25
$S_{4\rightarrow 0objects}$	6*0=0	1	0	0.5	0
Avg. **DCS** = 0.935; Avg. **VA** = 0; Avg. **OFR** = 0.49; Avg. **Acc.** = 0.46;					
Acc.: $Accuracy = (TP+TN)/(TP+TN+FP+FN)$; $Specificity = TN/(TN+FP)$; **N**: No. of trials; **OI**: Object Instances per each assembly state; **T**: Sum of total objects per assembly state over all experiments, **DCS**: TP/(TP+FN); **VA**: TN/(TN+FP); **OFR**: (DCS+VA)/2;					

TABLE III

EVALUATION TABLE FOR THE CEMOT SYSTEM

Assembly Steps	**CEMOT**				
	N*OI=T	**DCS**	**VA**	**OFR**	**Acc.**
$S_{0\rightarrow 4objects}$	10*4=40	0.95	NA	NA	0.91
$S_{1\rightarrow 3objects}$	8*3=24	0.91	0.80	0.855	0.88
$S_{2\rightarrow 2objects}$	6*2=12	1	1	1	1
$S_{3\rightarrow 1object}$	6*1=6	1	1	1	1
$S_{4\rightarrow 0objects}$	6*0=0	1	1	1	1
Avg. **DCS** = 0.93; Avg. **VA** = 0.95; Avg. **OFR** = 0.96; Avg. **Acc.** = 0.925;					

accuracy of the CEMOT system increases to 0.925, while the accuracy of SMOT amounts to 0.46.

A more detailed depiction of *State Verification* and *Action Verification* results of SMOT and CEMOT are described in the video[1] attachment. The video also demonstrates the ability of the CEMOT to deal with dynamic changes in the assembly process.

[1]Video link: https://youtu.be/8l4B0P8w8Go

TABLE IV

QUALITATIVE COMPARISON OF CEMOT SYSTEM

Approach	**Dynamic Changes**	**SI_MOT**	**MI_MOT**
Wan et al [7]	no	yes	no
Hossain et al [8]	partly	yes	no
Hamabe et al [9]	partly	yes	no
Savarimuthu et al [27]	yes	yes	no
Our Approach	yes	yes	yes
SI_MOT ability to handle single instances of multiple object types; **MI_MOT** ability to handle multiple instances of multiple object types; Dynamic changes including presence of humans - clutter, abrupt changes and occlusions			

To further clarify the contribution of our work, a qualitative comparison of the proposed approach against existing state of the art approaches is detailed in Table IV. The approaches chosen for comparison are those that deal with assembly processes in human robot collaboration scenarios and use *context* in some fashion to reinforce tracking/object recognition. The qualitative comparison evaluates the ability of the approaches in dealing with a) dynamic changes involving humans (clutter, abrupt changes and occlusions) in the interaction environment b) ability to track single instances of multiple object types and c) ability to track multiple instances of multiple object types.

VI. CONCLUSION

Assembly processes in industrial HRC often involve manipulation of several objects and demand coordination between the human agent and the robot. A major challenge in such dynamic environments is to make a qualified statement on the current execution state of the considered task. Especially, required states of objects of interest (e.g. spatial relations) are difficult to verify but play a major

role for reasoning on the current execution state. It is also of vital importance for such HRC system to determine if an executed action was indeed successful. Only relying on proprioception of the robot might not always convey the real consequence. Hence, other functional modalities available in the HRC system should be exploited for redundancy. Along these lines, we argue that the functional role of an object tracker should not just be localizing and tracking objects for manipulation. It should be extend to determine the current state and also verify the action performed.

For the object tracker to perform well in its functional role, relevant information about the current state of the AP (type of object, number of instances, etc) are of vital importance. However, extracting such information in a dynamic environment is a difficult challenge. In this work, we presented a framework that is capable of extracting such relevant information (*context*) for the tracker. The *context* extraction framework is then applied to a reconfigurable Context Enhanced Multi Object Tracking (CEMOT) system to help the tracker fulfill its functional role.

The novelty of CEMOT is that it applies 3D pose tracking of multiple objects in an assembly process involving human robot collaboration. The evaluation results are promising with an increase in overall accuracy from 49 to 96 percent, and motivate further integration of CEMOT with human action recognition for a robust activity recognition and as a result enable close cooperation between robot and human. In future work we plan to exploit the aspects of object anchoring with the CEMOT system. This could further improve CEMOT's performance in handling multiple instances of multiple objects.

ACKNOWLEDGMENT

This research is funded by the projects LERN4MRK (Austrian Ministry for Transport, Innovation and Technology), MMAssist_II (FFG, 858623), Smart Factory Lab and DigiManu (funded by the State of Upper Austria).

REFERENCES

[1] A. Bauer, D. Wollherr and M. Buss, Human–robot collaboration: a survey. International Journal of Humanoid Robotics, vol. 5, no. 1, pp. 47-66, 2008.

[2] T.B. Dinh, N. Vo, G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1177-1184, June 2011.

[3] Yang, M., Wu, Y., Hua, G.: Context-aware visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 7, pp. 1195-1209, July 2009 .

[4] S. Akkaladevi, M. Ankerl, C. Heindl, and A. Pichler, Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data, in Proc. IEEE International Conference on Robotics and Automation, pp. 5644-5649, 2016.

[5] A. Oliva and A. Torralba, The role of context in object recognition, Trends in cognitive sciences, Elsevier, vol. 11, pp. 520-527, 2007.

[6] L. Oksama and J. Hyönä, Is multiple object tracking carried out automatically by an early vision mechanism independent of higher-order cognition? An individual difference approach, Visual cognition, Taylor and Francis, vol. 11, pp. 631-671, 2004.

[7] W. Wan, F. Lu, Z.Wu, and K. Harada, Teaching robots to do object assembly using multi-modal 3d vision. Neurocomputing (2017).

[8] D. Hossain, G. Capi, M. Jindai, and S.I. Kaneko, Pick-place of dynamic objects by robot manipulator based on deep learning and easy user interface teaching systems. Industrial Robot: An International Journal, vol vol. 44, no. 1, pp. 11-20 (2017).

[9] T. Hamabe, H. Goto, and J. Miura, A programming by demonstration system for human-robot collaborative assembly tasks, in Proc. IEEE International Conference on. Robotics and Biomimetics (ROBIO), 2015.

[10] S. C. Akkaladevi, M. Plasch, C. Eitzinger and B. Rinner, Context Enhanced Multi-Object Tracker for Human Robot Collaboration, in Proc. 2017 ACM/IEEE International Conference on Human-Robot Interaction (late breaking reports),2017.

[11] L. Onofri, P. Soda, M. Pechenizkiy and G. Iannello, A survey on using domain and contextual knowledge for human activity recognition in video streams, Expert Systems with Applications, vol. 63, pp. 97-111, 2016.

[12] T. B. Dinh, N. Vo and G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1177-1184, 2011.

[13] Y. Zhu, N. M. Nayak and A. K. Roy-Chowdhury, Context-aware activity recognition and anomaly detection in video, IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 1, pp. 91-101, 2013.

[14] L. Snidaro, J. Garca and J. Llinas, Context-based information fusion: a survey and discussion, Information Fusion, vol. 25, pp. 16-31, 2015.

[15] E. Maggio and A. Cavallaro, Learning scene context for multiple object tracking, IEEE Transactions on Image Processing, vol. 18, no. 8, pp. 1873-1884, 2009.

[16] H. T. Nguyen, Q. Ji and A.W. Smeulders, Spatio-temporal context for robust multitarget tracking, IEEE transactions on pattern analysis and machine intelligence, vol. 29, no. 1, pp. 52-64, 2007.

[17] J. Amores, N. Sebe and P. Radeva, Context-based object-class recognition and retrieval by generalized correlograms, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp.1818-1833, 2007.

[18] D. Ognibene, E. Chinellato, M. Sarabia, and Y. Demiris, Contextual action recognition and target localization with an active allocation of attention on a humanoid robot, Bioinspiration and biomimetics, vol. 8, no. 3, 2013.

[19] D. Ognibene and G. Baldassare, Ecological active vision: Four bioinspired principles to integrate bottomup and adaptive topdown attention tested with a simple camera-arm robot, IEEE Transactions on Autonomous Mental Development, vol. 7, no. 1, pp. 3-25, 2015.

[20] S. C. Akkaladevi, M. Plasch, A. Pichler and B. Rinner, Human Robot Collaboration to Reach a Common Goal in an Assembly Process, in Proc. European Starting AI Researcher Symposium, vol. 284, pp. 3-14, 2016.

[21] M. Tenorth and M. Beetz, KnowRob - knowledge processing for autonomous personal robots, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4261-4266, 2009.

[22] P. Blackburn, J. Bos, and K.Striegnitz, Learn prolog now!, vol.7, College Publications, 2006.

[23] S. C. Akkaladevi and C. Heindl, Action recognition for human robot interaction in industrial applications, in Proc. IEEE International Conference on Computer Graphics, Vision and Information Security, pp. 94-99, 2015.

[24] S. C. Akkaladevi, M. Plasch, C. Eitzinger, S. C. Maddukuri and B. Rinner, Towards Learning to Handle Deviations Using User Preferences in a Human Robot Collaboration Scenario, in Proc. 8th International Conference on Intelligent Human Computer Interaction, Springer International Publishing, pp. 3-14, 2017.

[25] D. M. W. Powers, Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation, Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37-63, 2011.

[26] J. Johnson, A. Karpathy, and L. Fei-Fei, Densecap: Fully convolutional localization networks for dense captioning, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 4565-4574, 2016.

[27] T.R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Rossmann, D. Martínez, G. Alenyà, C. Torras, A. Ude, B. Nemec, A. Kramberger, F. Wörgötter, E. E. Aksoy, J. Papon, S. Haller, J. Piater, and N. Krüger, Teaching a robot the semantics of assembly tasks. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 5, pp. 670-692, 2018.