Automatic Interaction and Activity Recognition from Videos of Human Manual Demonstrations with Application to Anomaly Detection

Elena Merlo[1,2], Marta Lagomarsino[1,3], Edoardo Lamon[1], and Arash Ajoudani[1]

***Abstract*— This paper presents a new method to describe spatio-temporal relations between objects and hands, to recognize both interactions and activities within video demonstrations of manual tasks. The approach exploits Scene Graphs to extract key interaction features from image sequences, encoding at the same time motion patterns and context. Additionally, the method introduces an event-based automatic video segmentation and clustering, which allows to group similar events, detecting also on the fly if a monitored activity is executed correctly. The effectiveness of the approach was demonstrated in two multi-subject experiments, showing the ability to recognize and cluster hand-object and object-object interactions without prior knowledge of the activity, as well as matching the same activity performed by different subjects.**

I. INTRODUCTION

The comprehension of human activities enables machines to understand and interpret the visual information that they receive from the world around them and to make informed decisions based on what they see [1]. Many robotic applications also rely on video comprehension, such as autonomous navigation, interactions with objects, and human-robot interaction. In particular, in the latter, it is of paramount importance to be able to recognize human activities and predict their outcomes. Examples of these applications can be found in scenarios where robots provide assistance in Activities of Daily Living (ADL) [2] or collaborate with humans in industrial settings to achieve a common goal [3]. Moreover, by understanding a task demonstrated by humans, robots could learn the task structure and replicate the activities [4]. Both domestic and industrial activities are characterized by the prominent presence of manual tasks. However, being able to describe in detail human manipulation activities is a challenging problem. In particular, among the open research questions, the first complexity is represented by the selection and extraction of features [5], [6] which should provide comprehensive, yet compact, scene descriptions, avoiding data overfitting, and the related video segmentation problem [7], [8], which allows to face the recognition problem at a more granular level. In the context of human manual activities analysis, researchers have employed various approaches depending on the specific application they intended to achieve. Some studies prioritized features related to the involved objects [9], [10], while others concentrated on describing trajectories of human arms [11], [12]. However, to provide a general description, exhaustive approaches have looked at a variety of factors, for example the motion of the hand, its relative location with respect to objects [13], the location of the objects, the distance between objects, and the semantic relationships between the hands and objects [4], [14]. However, what is missing in these approaches is an explicit modeling and extraction of the motion pattern during a hand-object or object-object interaction. Although they consider the motion of hands and objects and their spatial relationships, they do not explicitly capture detailed information about the motion patterns during an interaction, such as the velocity and direction of the objects involved.

To provide a more detailed understanding of the interaction flows, we propose a method to include all relevant features through a scene encoder and to automatically segment activities hierarchically. Our approach overcomes the State-of-the-Art (SoA) by describing objects' spatial relationships and integrating information about the motion patterns during interactions. This way, unlike the above-mentioned literature studies, we will be able to distinguish different manners of executing the same activity, or a successful execution from a warped one, which can be crucial for accurately mapping a recognized interaction to the corresponding robot behavior. Moreover, our encoding can separate contextual information from motion information, enabling us to recognize analogous motion patterns with different objects as similar interactions. For example, assembling and disassembling might involve the same objects but opposite motions. Therefore, the proposed method aims to recognize individual activities by identifying and grouping time sequences of frames corresponding to interactions. To achieve this result, we collect a dataset of human manipulation demonstrations and extract features such as the position, velocity, and acceleration of the objects and the hands involved in the interaction, as well as the orientation, shape, and distance of the objects. Then, given the definition of our taxonomy, we define a novel method to segment and compare these time-series of features, to capture the temporal patterns of the interactions, and to cluster similar interactions altogether. At the same time, by comparing an online execution of an activity with the learned ones through the proposed metrics, the method could also be used to detect potential anomalies in the activity progress. By means of such unsupervised approach, it is possible to extract patterns and similarities from the data itself, without relying on pre-learned activity models or labels. This can lead to prompt and flexible application of the learning strategy, as it is possible to identify commonalities

[1] Human-Robot Interfaces and Interaction Laboratory, Istituto Italiano di Tecnologia, Genoa, Italy. elena.merlo@iit.it

[2] Dept. of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genoa, Genoa, Italy.

[3] Dept. of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy.

between different scenarios at a lower layer (the interaction layer) and easily generalize knowledge to new situations. The method has been tested in two multi-subject experiments, that aimed at evaluating i) the capability of the method in recognizing similarities in activities executed by different subjects and ii) the potential of the method in detecting online activity anomalies given a learned activity representation (see multimedia attachment[1]). The results demonstrate the effectiveness of our proposed approach in recognizing and clustering hand-object and object-object interactions and in matching the same activity executed by different subjects.

II. EVENT-BASED TAXONOMY

In this study, we focus on the analysis of manual activities that involve the manipulation of objects through human hands. Henceforth, we will refer to both objects and human hands as *video objects*, which are, using a computer vision terminology, specific entities or regions of interest within a video sequence that a computer system is able to identify and track over time.

In our approach, we examine the spatio-temporal relationships between video objects by analyzing the content of smaller video segments obtained by video segmentation. By segmenting the video into atomic units and then grouping these interactions together according to logic rules, it is possible to describe and recognize the activity occurring in the video. In other words, we are aiming to understand higher-level activities starting from descriptions of low-level interactions, adopting a bottom-up hierarchical approach. To further clarify the problem and the goal of the work, we provide an event-based taxonomy, which extends the one proposed by *Fu et al.* [8] to define each different event:

- an **Elementary Reaction Unit** (ERU) is a set of consecutive frames within which video objects have a specific spatio-temporal relationship. The onset of a new ERU is due to a change in the video objects' relationship;
- an **Interaction Unit** (IU) is a time-ordered sequence of ERUs that involve the same video objects. By grouping together the ERUs, we are able to capture all of the changes that occur in the spatio-temporal relationship between the video objects;
- an **activity** is a time-ordered sequence of IUs. The IUs within an activity are logically connected, with the successful completion of one IU leading to the start of the next. When two IUs are not connected logically it means that the activity is changed;
- a **job** is a collection of activities which share an overall common objective.

In particular, the definitions of ERU and IU are based on [8]. However, the taxonomy lacks in capturing the overall structure and context of the interactions. That is why we introduced the activity and job layers, which allow us to group hierarchically IUs into broader, more meaningful units.

III. METHODOLOGY

The framework depicted in Figure 1 illustrates the various stages of the proposed approach. The input of the framework is a video demonstration of the manipulation task. The perception module is responsible for extracting the 3D hands' landmarks and the 3D pose of the objects in the scene. It should be noted that the presented method is not limited to a particular technique to detect hand and object movements. Instead, it can accommodate various approaches, including inertial motion capture, marker-based motion capture with multiple cameras, or RGBD-based markerless motion capture. Additionally, information on the objects type and interaction points are required as input. These points are chosen based on the object properties such as shape, size, and affordance, and correspond to the suitable grasp frames for object interactions with hands or other objects [15]. Similarly, providing a technique to automatically retrieve the object interaction points is out of the scope of the manuscript. However, SoA techniques in robotic manipulation and grasping can be employed [16]. Once extracted, the perception module outputs are organized per frame into a Scene Graph (SG) structure [17], [18] that portrays the spatial semantic relationships among the video objects in the scene. This representation, enriched with local temporal features, such as video objects' relative accelerations and velocities, could provide a detailed and sensitive-to-variation scene description. In this way, specific changes in the feature space represent variations in the scene content, leading to the segmentation of the video into ERUs, IUs, or activities. Once the video has been segmented, it is possible to recognize and compare different patterns. In particular, in this manuscript, we will focus on grouping IUs using clustering techniques, such as centroid- or distribution-based clustering [19], [20]. This involves comparing the IUs based on similarity measurements and grouping together the similar ones. The feature set consists of two components. The first component encodes motion features at the interaction level, regardless of which objects are involved. The second one encodes contextual information, such as the identity of the objects involved in the interaction. When clustering motion features, we obtain clusters of context-free IUs that are characterized by similar motion patterns. Instead, when clustering contextual information, we group IUs that involve the same objects being handled. The latter allows us to identify interactions based on context that are equal in terms of the specific objects involved, even if their motion features differ. However, to discriminate IUs, both feature components are necessary. Therefore, an ensemble-like approach is used, where the results of each independent clustering are combined into a single cluster.

A. Scene Encoding

By processing the perception data, each frame representing a scene is mapped into a SG. A SG G is defined as the tuple $G = (V, R, E)$. $V = \{v_1, \ldots, v_{|V|}\}$ is the set of video

[1]The video can also be found at youtu.be/Ftu_EHAtH4k.

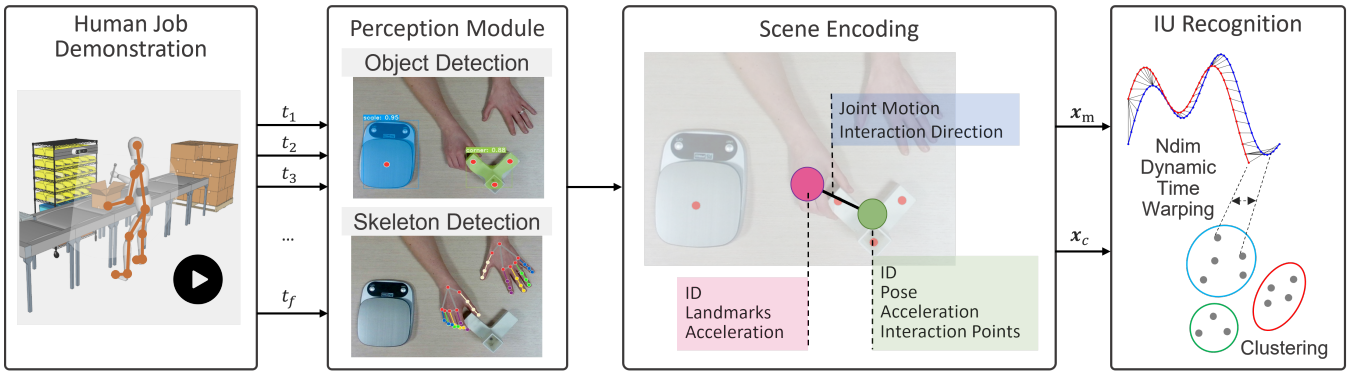

Fig. 1. Overview of the overall framework. Given a video demonstration of a human job, the perception module detects and extracts the 3D objects poses and hands landmarks positions. This data is organized per frame into a Scene Graph structure, where objects and hands are the nodes, and edges represent interactions between the connected nodes. The scene's content is encoded into a feature space, capturing both motion $\boldsymbol{x}_{\mathrm{m}}$ and context $\boldsymbol{x}_{\mathrm{c}}$ at the interaction level. Changes in the feature space indicate variations in the scene, leading to the segmentation of the video into Interaction Units (IU)s. IUs can be recognized and compared using clustering techniques to group together similar ones.

objects which are interacting with each other (foreground). We discard not interacting objects since they are considered not relevant in the scene description (background). Each object is represented as $v_{\mathrm{i}} = (c_{\mathrm{i}}, A_{\mathrm{i}})$, where c_{i} and A_{i} respectively indicate the category and attributes of the video object. R represents a set of relationships between the nodes, while E denotes the edges between the two interacting video objects [18]. Furthermore, in order to describe the interactions for each hand separately as in [21], we will generate n SGs if there are n hands interacting with the same object. This allows us to capture the unique interactions of each hand with the objects.

1) Scene Graph nodes: Each node $v \in V$ in a scene graph represents a video object, with different attributes depending on the type. The *i-th* hand node is represented by $v_{\mathrm{h_i}} = (c_{\mathrm{h_i}},\ A_{\mathrm{h_i}})$, with $c_{\mathrm{h_i}} = ID_{\mathrm{h_i}}$, and $A_{\mathrm{h_i}} = (LM_{\mathrm{h_i}},\ \boldsymbol{a}_{\mathrm{h_i}})$, where ID_{h} is the hand identity, $\boldsymbol{a}_{\mathrm{h_i}} \in \mathbb{R}^3$ is the Cartesian acceleration of the hand, and $LM_{\mathrm{h_i}}$ is the set of hand landmarks which represent key locations such as fingertips, knuckles, and wrist. $\boldsymbol{a}_{\mathrm{h_i}}$ is the hand acceleration, measured at the the middle finger knuckle. A subset of $LM_{\mathrm{h_i}}$, in particular the landmarks corresponding to the fingertips of the three middle fingers, represents the hand interaction points $IP_{\mathrm{h_i}}$. Instead, the *j-th* object node is represented by $v_{\mathrm{o_j}} = (c_{\mathrm{o_j}},\ A_{\mathrm{o_j}})$, with $c_{\mathrm{o_j}} = ID_{\mathrm{o_j}}$, and $A_{\mathrm{o_j}} = (\boldsymbol{p}_{\mathrm{o_j}},\ \boldsymbol{\phi}_{\mathrm{o_j}},\ \boldsymbol{a}_{\mathrm{o_j}},\ IP_{\mathrm{o_j}})$, where $ID_{\mathrm{o_j}}$ is the object identity, $\boldsymbol{p}_{\mathrm{o_j}}$ is the position (based on the object centroid), $\boldsymbol{\phi}_{\mathrm{o_j}}$ is the orientation, $\boldsymbol{a}_{\mathrm{o_j}} \in \mathbb{R}^3$ is the Cartesian acceleration, and $IP_{\mathrm{o_j}}$ is the set of the interaction points position, computed with respect to $\boldsymbol{p}_{\mathrm{o_j}}$.

2) Scene Graph edges: An edge $e_{\mathrm{i,j}} \in E$ connects two video objects v_{i} and v_{j} if and only if

$$d_{\mathrm{i,j}} = dist(IP_{v_{\mathrm{i}}}^{\mathrm{k}},\ IP_{v_{\mathrm{j}}}^{\mathrm{q}}) \approx 0, \quad \forall\, k,q \in [1,\ |IP_v|],\ k \neq q,$$

where $dist(\cdot,\cdot)$ is the Euclidean distance. In other words, we determine whether there is an interaction between two video objects if the minimum Euclidean distance between each pair of their respective interaction points $IP_{v_{\mathrm{i}}}$ and $IP_{v_{\mathrm{j}}}$ is reasonably small as in [4]. Each $e_{\mathrm{i,j}} \in E$ is described by a relationship between the connected nodes v_{i} and v_{j}, $r_{\mathrm{i}\to\mathrm{j}} \in R$. Each $r_{\mathrm{i}\to\mathrm{j}} \in R$ includes three attributes:

- the *distance* $d_{\mathrm{i,j}}$ between the two objects v_{i} and v_{j};
- the *joint motion*: whether v_{i} and v_{j} are moving jointly;
- the *relative motion direction*, expressing whether v_{i} is moving towards or away from v_{j}.

To determine whether the two sufficiently close video objects are moving jointly, we compare their acceleration signs. If $sgn(\boldsymbol{a}_{\mathrm{v_i}})$ and $sgn(\boldsymbol{a}_{\mathrm{v_j}})$ are concordant, we can conclude that v_{i} and v_{j} are moving together. In the case where a hand and an object are jointly moving, we define a *in-hand* relation between them, i.e., we assume that the hand is holding the object.

In addition, the interaction direction is obtained by projecting the velocity vector of the moving object v_{i} onto the frame $T_{v_{\mathrm{j}}} = [\boldsymbol{p}_{v_{\mathrm{j}}}, \boldsymbol{\phi}_{v_{\mathrm{j}}}]$ of the stationary object v_{j}; the resulting vector is mapped in spherical coordinates, i.e., elevation $\theta_{\mathrm{i,j}}$ and azimuth $\varphi_{\mathrm{i,j}}$ angles and radius $\rho_{\mathrm{i,j}}$, and then discretized: $\theta_{\mathrm{i,j}}^{\mathrm{Q}} = Q(\theta_{\mathrm{i,j}}), \quad \varphi_{\mathrm{i,j}}^{\mathrm{Q}} = Q(\varphi_{\mathrm{i,j}})$. As a result, $\theta_{\mathrm{i,j}}^{\mathrm{Q}}$ and $\varphi \mathrm{i,j}^{\mathrm{Q}}$ describe the interaction direction of video objects using a finite number of quantized values. A strong point of this representation is the space invariance allowing us to encode in the same way interactions that involve similar relative approaching directions regardless of the video objects' absolute poses.

3) Feature Couple: The scene representation provided by the SG can be further reduced by means of a feature couple, denoted as $X = (\boldsymbol{x}_m, \boldsymbol{x}_c)$, where $\boldsymbol{x}_{\mathrm{m}}$ conveys semantic motion information (*motion features*), while $\boldsymbol{x}_{\mathrm{c}}$ about the video objects IDs (*context features*). To achieve such a compact representation, we select $e_{\mathrm{h_i,o_j}}, e_{\mathrm{o_f,o_g}} \in E$ respectively describing the hand-object and object-object interactions with the smallest d. As a result, X conveys the description of at most one object-object interaction and one hand-object interaction, independently from the number of nodes $|V|$.

$$\boldsymbol{x}_{\mathrm{m}} = \begin{bmatrix} \alpha_{\mathrm{h_i}} & \theta_{\mathrm{o_f,o_g}}^{\mathrm{Q}} & \varphi_{\mathrm{o_f,o_g}}^{\mathrm{Q}} & \theta_{\mathrm{h_i,o_j}}^{\mathrm{Q}} & \varphi_{\mathrm{h_i,o_j}}^{\mathrm{Q}} & jm_{\mathrm{o_f,o_g}} & jm_{\mathrm{h_i,o_j}} \end{bmatrix}$$
$$\boldsymbol{x}_{\mathrm{c}} = \begin{bmatrix} ID_{\mathrm{o_f}} & ID_{\mathrm{o_g}} & ID_{\mathrm{h_i}} & ID_{\mathrm{o_j}} \end{bmatrix}$$

In $\boldsymbol{x}_{\mathrm{m}}$, the first feature $\alpha_{\mathrm{h_i}}$ represents the hand acceleration $\boldsymbol{a}_{\mathrm{h_i}}$ categorized into three states: accelerating, decelerating, or still. The spherical angles $\theta_{\mathrm{o_f,o_g}}^{\mathrm{Q}}$, $\varphi_{\mathrm{o_f,o_g}}^{\mathrm{Q}}$ and $\theta_{\mathrm{h_i,o_j}}^{\mathrm{Q}}$, $\varphi_{\mathrm{h_i,o_j}}^{\mathrm{Q}}$

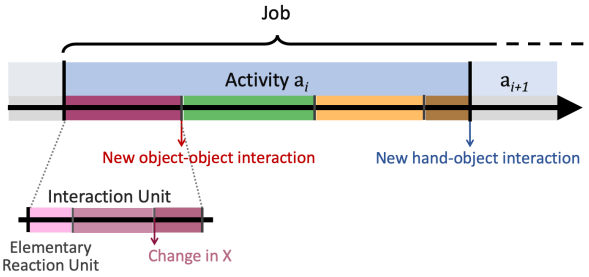

Fig. 2. Conceptual illustration of the proposed taxonomy and automatic video segmentation criteria.

represent the direction of the interaction between objects and the interaction between objects with the hand, respectively. The sixth and seventh binary features encode the object-object $jm_{\mathrm{o_f,o_g}}$ and hand-object $jm_{\mathrm{h_i,o_j}}$ joint motions. On the other hand, $\boldsymbol{x}_{\mathrm{c}}$ contains the ID of the interacting video objects. More specifically, the first two features are the IDs of the two objects involved in the object-object interaction, i.e. $ID_{\mathrm{o_f}}$ and $ID_{\mathrm{o_g}}$. The last two features instead specify the IDs of the hand (i.e. $ID_{\mathrm{h_i}}$) and the object (i.e. $ID_{\mathrm{o_j}}$) in the hand-object interaction.

B. Event-Based Video Segmentation

Using the proposed encoding from a time series of video frames, we obtain a time series of X. The video segmentation can then be easily automatized by examining specific changes in the feature values. In this paper, we propose the following rules for the automatic segmentation of ERUs, IUs, and activities (see Figure 2):

- a new ERU begins when any change occurs in X;
- a new IU is initiated when there is a change in $\boldsymbol{x}_{\mathrm{c}}$. Specifically, if at least one video object starts or stops interacting with others, a new IU arises;
- a new activity starts with a change in the last feature of $\boldsymbol{x}_{\mathrm{c}}$. This implies that a new activity arises when the hand interacts with a new object.

Let's explain the segmentation strategy with an example. Consider the job of filling a box with five tools. The proposed segmentation would return five similar activities characterized by the hand interacting with a specific object and would involve picking up a tool and placing it inside the box. Each activity would be, in turn, composed of the following four IUs: (i) the human hand grasping the tool from the storage area (involving hand-tool and tool-storage area interactions), (ii) the hand holding the tool in the air far from the storage area (involving hand-tool interaction but no object-object interaction), (iii) placing the tool in the box (involving hand-tool and tool-box interactions), and (iv) the tool becoming integral with the box (involving no hand-object interaction but tool-box interaction). A new activity starts when a new tool is grasped, and the four IUs are repeated.

C. Similarity Measures and Clustering

Following the procedure above, motion features $\boldsymbol{x}_{\mathrm{m}}$ and context features $\boldsymbol{x}_{\mathrm{c}}$ can be associated with each video frame, enabling the description of each IU both in terms of motion and of the objects involved in the interaction. In this section, we describe the metrics used to find similar motion patterns among IUs, and discern IUs that involve the same objects. Moreover, we propose a machine learning approach to automatically group IUs describing similar interactions without previous knowledge about the accomplished activities.

1) Similarity between IUs motion patterns: To measure the similarity between IUs in terms of motion, we utilize the multi-dimensional Dynamic Time Warping (DTW). DTW is a widely-used distance measure for time series data, and it allows us to compare sequences of different lengths by warping and stretching the time axis. The resulting distance reflects the discrepancy between two context-free IUs, i.e., how different two IUs are based on their motion patterns. The lower the distance, the higher the similarity. It should be noticed that the approach can be customized and scaled to specific task requirements. Indeed, by applying a binary mask on $\boldsymbol{x}_{\mathrm{m}}$, it is possible to feed the DTW with a subset of $\boldsymbol{x}_{\mathrm{m}}$ including only the desired features. For instance, if our objective is to identify the bread slicing activity, regardless of how the slices are performed, we could ignore the information related to the object-object interaction direction by disregarding features $\theta^{\mathrm{Q}}_{\mathrm{o_f,o_g}}$ and $\varphi^{\mathrm{Q}}_{\mathrm{o_f,o_g}}$. In doing so, both horizontal and vertical slicing would be assigned to the same cluster. While the distance provided by DTW is a promising candidate measure of the similarity between IUs, determining the condition under which two video segments should be considered as instances of the same interaction is not trivial. To overcome this issue, we exploit the K-means clustering algorithm [19], where a suitable value of k can be deduced with the elbow method applied to the trend of the Within-Cluster Sum of Squares (WCSS) over k.

2) Similarity between IUs contexts: By definition, each IU is characterized by the same video objects in interaction, thus by a constant $\boldsymbol{x}_{\mathrm{c}}$. Hence, two distinct $\boldsymbol{x}_{\mathrm{c}}$ are representative of different IUs. This means that two IUs context are similar (actually identical) if their distance is smaller than 1 for each possible distance metric. Therefore, to cluster all the IUs involving the same video objects' interactions, we utilize DBSCAN [20] with Euclidean distance and $\varepsilon = 1$ as the maximum cluster distance.

By combining the two clusterings, we can detect IUs that present similarities both in terms of motion and context.

D. Anomaly Detection in Activity Execution

The proposed method to segment and distinguish activities can be particularly useful in application where the ability to identify and respond to plan deviations on the fly is critical. For instance in the anomaly detection in human job executions, a prompt anomaly identification could trigger alerts or corrective actions.

We assume that an activity is correctly performed when all IUs are properly executed and in the correct time order. The algorithm capitalizes on three lists J, L, and C. The first one contains all the nominal activities required to complete the selected job. The second one comprehends the activities discarded as non-candidates. The latter instead includes the

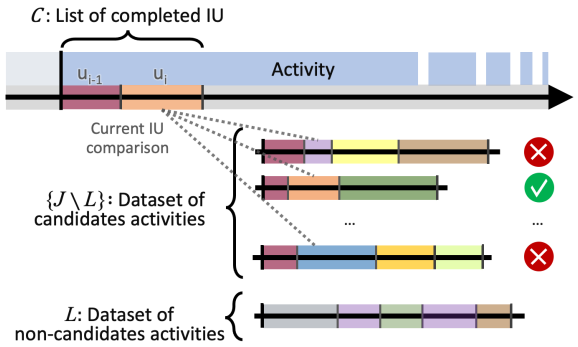

Fig. 3. Selection of candidates in anomaly detection algorithm. The first completed IU was compared with the first IUs of all the candidate activities in J. This comparison resulted in a division between candidate $\{J \setminus L\}$ and non-candidate L activities. Second completed IU is compared with the second IU of the remaining candidate activities. Once each IU is completed, it is added to the list C. This process continues until the end of the activity, when the list C is checked if it perfectly matches with an activity in $\{J \setminus L\}$.

completed IUs. L and C are initially empty. During the job execution, the recorded scenes are encoded and segmented using the above-mentioned procedure. Once a new IU u_{i+1} starts, the last completed one, u_i, is added to list C and is compared to all the *i-th* IUs of each activity in the list $\{J \setminus L\}$ in terms of context and motion. First, it is checked if u_i and the candidate ${}^{a}u_i$ have identical context (i.e., $\boldsymbol{x}_{c,i}$==${}^{a}\boldsymbol{x}_{c,i}$). Then, the DTW algorithm is used for the comparison of the motion features ($DTW(\boldsymbol{x}_{m,i}, {}^{a}\boldsymbol{x}_{m,i})$). If $d > d_{th,i}$, that activity is added to the list L, which means that it is no longer a candidate activity. Otherwise, the activity remains a candidate. If the current IU is not included in any of the activities belonging to the list $\{J \setminus L\}$, an alert will be generated to indicate the occurrence of an anomaly. The process is repeated for each IU until the current activity ends. Figure 3 shows graphically the selection of candidates. Once the activity ends, the algorithm checks if the sequence of the executed IUs, stored in list C, corresponds to the series of IUs of one of the candidate activities remaining in $\{J \setminus L\}$. At this point, all the activities contained in $\{J \setminus L\}$ have the same time-ordered sequence of i IUs. If a match is found, it means that the activity was executed correctly. Otherwise, the activity was not completed and the algorithm reports the anomaly (see Algorithm 1).

Algorithm 1 Anomaly Detection in Activity Execution

```
J ← list of the nominal activities of a job
d_th ← distance thresholds for each IU
C ← {}                                   ▷ List of completed IUs
L ← {}                                   ▷ List of non-candidate activities
i ← 1
while job is not finished do
    activity_in_J ← True
    while activity is not finished do
        if u_i is finished then
            u_i ← just completed IU
            x_m,i, x_c,i ← just completed IU features
            Add u_i to C
            for each activity a in {J \ L} do
                                  ▷ Computing similarity with reference IU
                ^a x_m,i, ^a x_c,i ← i-th IU of a features
                if x_c,i == ^a x_c,i then
                    d ← DTW(x_m,i, ^a x_m,i)
                    if d > d_th,i then          ▷ Non-similar motion
                        Add a to L   ▷ a is non-candidate activity
                    end if
                else                            ▷ Non-similar context
                    Add a to L       ▷ a is non-candidate activity
                end if
            end for
            if {J \ L} == ∅ then
                activity_in_J ← False   ▷ No such activity in set J
            end if
            i ← i + 1
        end if
    end while
    activity_is_correct ← False
    if activity_in_J then
        for a* in {J \ L} do
            if a* ≡ C then              ▷ Activity executed correctly
                L ← {}, C ← {}
                activity_is_correct ← True
                break
            end if
        end for
        if not activity_is_correct then     ▷ Activity not completed
        end if
    end if
end while
```

IV. EXPERIMENTS

The evaluation of the method consisted of two experiments. In the first one, we validated the proposed video segmentation and the IUs similarity recognition, while in the second experiment, we tested our anomaly detection algorithm on the activity execution. The experimental setup envisioned an RGB camera (IntelRealSense Camera) mounted in top shot (bird's eye) view, and the image plane was aligned with the working plane (i.e., the tabletop of the workbench). This way, we could exploit ArUco markers as 3D object detection method. However, the same method did not give satisfactory results in hand detection due to the different configurations that the hand can have during the manipulation task, eventually hiding the marker, and since the marker hindered the natural movements of participants. For this reason, we utilized the MediaPipe Hand Detector, which features 21 hand landmarks. For simplicity, we detected only the right hand and evaluated the method in 2D (in the image plane, which is aligned with the workbench tabletop). As a result, feature φ was not taken into account. The architecture has been developed in Python, on Ubuntu 20.04 and ROS Noetic, exploiting the DTAIdistance library [22] for the multi-dimensional DTW and the k-means clustering and Scikit-Learn library [23] for the DBSCAN with Euclidean distance.

A. Activity Recognition Validation (Exp 1)

In the first experiment, we asked $N_{sub} = 10$ right-handed subjects, 7 males and 3 females (25.6 ± 1.2 years old), to perform a set of five activities for $N_{rep} = 4$ repetitions. These activities involved the following objects: an aluminum profile, a corner joint, a meter, a box, a polisher, and a black brick (shown in Figure 4). The features X associated with

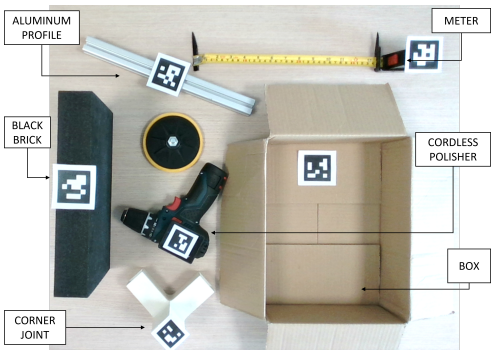

Fig. 4. Objects used in the experiments.

each activity, normalized in $[0,1]$, were filtered using the opening-closing filter [24]. The activities and IUs segmented by the algorithm are listed below (the semantic labels are not used by the algorithm and are reported only to clarify the experiment):

I) *Boxing*: grasp the profile, put the profile inside the box, leave the profile inside the box;
II) *Measuring*: grasp the profile, put the profile close to the meter, leave the profile close to the meter;
III) *Assembly*: grasp the profile, interlock the profile and the corner joint, leave the complete assembly;
IV) *Disassembly*: profile and corner joint already interlocked, disassemble the profile and the corner joint, move away from the corner while holding the profile;
V) *Polishing*: grasp the polisher, polish the black brick with back and forth motions, move away from the black brick surface while holding the polisher.

To further evaluate the similarity between IUs, we selected a subset of size $N_{\text{IU}} = 6$ of representative IUs:

1) grasp the profile, from *measuring*;
2) put the profile inside the box, from *boxing*;
3) put the profile close to the meter, from *measuring*;
4) interlock the profile and the corner joint, from *assembly*;
5) disassemble the profile and the corner joint, from *disassembly*;
6) polish the black brick with back and forth motions, from *polishing*.

We initially analyzed the IUs extracted from the $N_{\text{rep}} = 4$ repetitions of each activity by a single subject, computing similarities in terms of motion features and context features. The similarities between the context-free IUs were evaluated using DTW and a confidence matrix of size $(N_{\text{IU}} \times N_{\text{rep}})x(N_{\text{IU}} \times N_{\text{rep}})$ was generated to show the distances between each couple of IUs. Figure 5 shows the Single subject Confidence Matrix (SCM) for subject 5, which reports similarities between context-free IUs. The distances obtained from DTW were normalized to a range between 0 and 1, and dark patches indicate lower distance, hence higher similarity between IUs.

To evaluate instead the similarity of the same IU performed by different subjects, we conducted a multi-subject analysis by computing the distances between all the repetitions across all the participants. We filled the corresponding

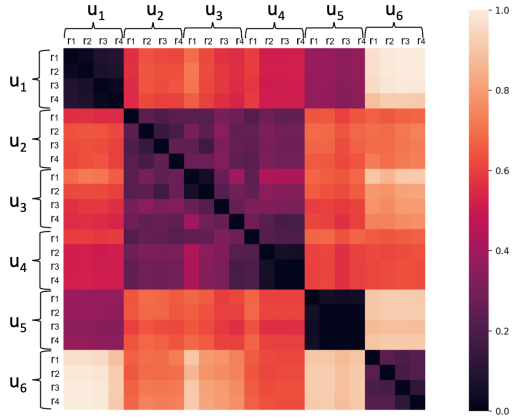

Fig. 5. Single subject Confidence Matrix resulting from the comparison of context-free IUs. The IUs are respectively: u_1 grasp the profile, u_2 put the profile in the box, u_3 measure the profile, u_4 assemble the profile and corner, u_5 disassemble the profile and corner, u_6 polish the black brick surface. r_{i} corresponds to the *i-th* repetition. A darker color in the matrix indicates a higher similarity between the motion patterns of the IUs being compared.

confidence matrix Multi subjects Confidence Matrix (MCM) with size $(N_{\text{IU}} \times N_{\text{sub}})x(N_{\text{IU}} \times N_{\text{sub}})$ (see Figure 6). Each element is computed as:

$$MCM(\text{a},\text{x},\text{b},\text{y}) = \frac{1}{N_{\text{rep}}^2} \sum_{m=1}^{N_{\text{rep}}} \sum_{n=1}^{N_{\text{rep}}} DTW((u_{\text{a}}, s_{\text{x}}, r_{\text{m}}), (u_{\text{b}}, s_{\text{y}}, r_{\text{n}}))$$

where a,b $\in [1, N_{\text{IU}}]$ x,y $\in [1, N_{\text{sub}}]$. u_{a} and u_{b} identify the couple of IUs we are comparing, while s_{x} and s_{y} denote the subjects. In other words, each element of MCM represents the distance between the average performance of all the N_{rep} repetitions of IU u_{a} for subject s_{x} and the average performance of all the N_{rep} repetitions of IU u_{b} for subject s_{y}. Distances were then normalized in $[0,1]$.

We further analyzed the results by clustering all executions of the IUs by all subjects using k-means with DTW as distance metric. To determine the optimal number of clusters, we ran the algorithm 10 times for $k \in [1,10]$ and selected the minimum WCSS for each k. The best value of $k = 4$ was given by the elbow method (see Figure 7 (left)). In particular, IUs of the *assembly*, *measuring*, and *boxing* activities were grouped together in the same cluster (see Figure 7 (top-right)). At the same time, to compare IUs context, we used the DBSCAN algorithm with Euclidean distance and maximum cluster distance $\varepsilon = 1$. In this case, the clustering algorithm identified a total of $k = 5$ clusters, one for each type of IU except for those from *assembly* and *disassembly* activities, which were grouped into a single cluster (see Figure 7 (bottom-right)).

Besides, we conducted an additional analysis to verify the robustness of the detection with respect to variations of the absolute poses of the involved video objects within the same IUs. In particular, we asked the same participants to perform a *drilling* activity where the IUs included: 1) grasp the drill, 2) drill the black brick for 5 seconds, and 3) move away from the surface while holding the drill. This activity was carried out while changing the absolute position of the drill and the black brick in three different configurations (C1, C2, C3), as illustrated in Figure 8 (left), for a total of 30 activities

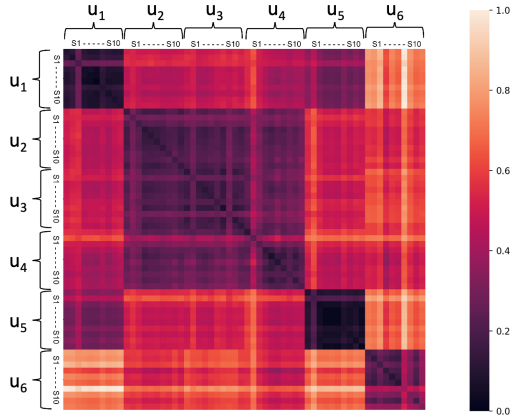

Fig. 6. Multi-subject Confidence Matrix resulting from the comparison of context-free IUs. The IUs are respectively: u_1 grasp the profile, u_2 put the profile in the box, u_3 measure the profile, u_4 assemble the profile and corner, u_5 disassemble the profile and corner, u_6 polish the black brick surface. $s_1 \ldots s_{10}$ corresponds to the subject number, which ranges from 1 to 10. A darker color in the matrix indicates a higher similarity between the motion patterns of the IUs being compared.

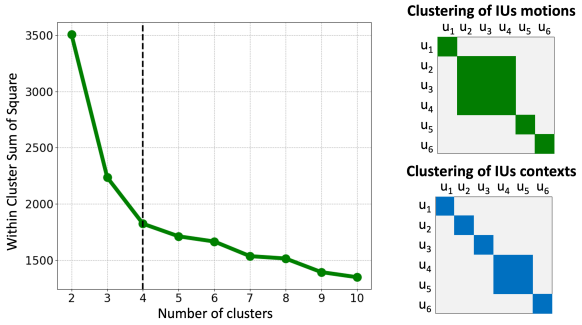

Fig. 7. Results of clustering IUs of multi-subject experiment. K-means clustering is used for clustering IUs motion patterns. The best k value $k=4$ is found with the elbow method (left). Results show u_2, u_3, and u_4 as a unique cluster (top-right). DBSCAN is used for clustering IUs contexts and it results in 5 clusters, one for each IU except for u_4 and u_5, which are grouped together in a single cluster (bottom-right).

executed. In Figure 8 (right) we reported the confidence matrix of the second IU.

B. Anomaly Detection Algorithm Validation (Exp 2)

Our second experiment focused on identifying anomalies in a job that consisted of two distinct activities: I) polishing the black brick surface, and II) measuring its thickness. Activity I) consisted of three IUs: 1) grasp the polisher; 2) polish the brick surface employing back and forth motions; 3) move away from the surface and release the polisher. Activity II) involved two IUs: 1) grasp the brick; 2) place the brick close to the meter.

We asked $N_{\text{sub}} = 7$ right-handed subjects, 6 males and 1 female (25.7 ± 1.1 years old) to perform the job correctly 3 times. Using the procedure described in subsection III-B, we automatically segmented each of the filtered $N_{\text{correct}} = N_{\text{sub}} \times 3 = 21$ job executions into $N_{\text{IU}} = 5$ IUs. The algorithm requires a nominal execution of the job and the distance threshold vector $\boldsymbol{d}_{\text{th}}$. The first was obtained by calculating the barycenter b_{u_i} of each IU, $i \in [1, N_{\text{IU}}]$. The latter corresponds to $d_{\text{th,i}} = \mu_{\text{u}_i} + 2\sigma_{\text{u}_i}$, where μ_{u_i} and σ_{u_i} are mean and standard

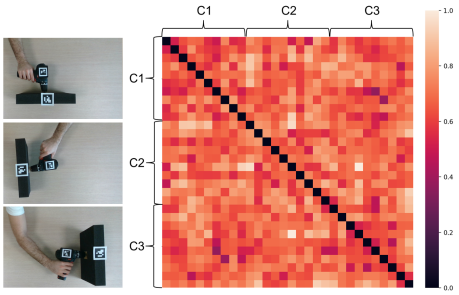

Fig. 8. Drilling activity being performed in three different configurations labeled as C1, C2, and C3 (left). Confidence matrix obtained by calculating the distance between all the IUs from each configuration (right).

TABLE I

CONFUSION MATRIX OF ANOMALY DETECTION

IU level		**Predicted**	
		Negative	Positive
Actual	Negative	858	52
	Positive	0	140
Accuracy = $95.0\% \pm 2.7\%$			

Activity level		**Predicted**	
		Negative	Positive
Actual	Negative	252	44
	Positive	0	124
Accuracy = $89.5\% \pm 5.1\%$			

deviation of DTW distance on the motion features of each IU from its corresponding barycenter. Finally, the nominal activities generated by the sequence of b_{u_i} are stored in J. Subsequently, we asked each subject to repeat two flawed job executions (J_1 and J_2). In J_1, we instructed the subjects to fail the second IU of activity I) by stopping halfway the polishing, while in J_2, we asked them to fail the second IU of activity II) by not measuring the brick.

The accuracy of the anomaly detection algorithm presented in Algorithm 1 was evaluated through 10-fold cross-validation. In each iteration, we randomly divided the set of correct executions, which contains $N_{\text{correct}} = 21$ samples, into a training set and a test set. The training set contained $N_{\text{training}} = 14$ randomly selected samples, while the test set contained the remaining $N_{\text{test}} = 7$ samples. At each iteration, the N_{training} were used for retrieving b_{u_i} and $d_{\text{th,i}}$ for each IU, while N_{test} along with the J_1 and J_2 executions from all the participants were given as input of the anomaly detection algorithm. We expected that during the analysis of the N_{test} executions, all IUs would be recognized as correctly performed, while the analysis of J_1 and J_2 should reveal errors in the second IU of activity I) and in the second IU of activity II), respectively. The IU and activity accuracy results are shown in the confusion matrix in Table I.

V. DISCUSSION AND CONCLUSIONS

In this paper, we proposed a bottom-up approach for recognizing activities by analyzing object-object and hand-object interactions in terms of motion and context information. Experiments in subsection IV-A demonstrated the capability of the framework to identify and group similar context-free interactions. This indicates that our scene encoding and features-based representation succeeded in comprehensively describing manual activities and identifying video

objects interaction' changes during the execution. Strong points of our encoding include the automatic segmentation of activities and IUs, as well as the possibility of separating contextual information from motion information. Considering the motion features exclusively, we can recognize IUs that involve similar motion patterns. Within our experiments, DTW identifies similarities between *boxing*, *measuring*, and *assembly* (see SCM in Figure 5). This outcome is motivated by the fact that the IUs mentioned above share (i) the hand holding of the profile, (ii) the approaching of a tool (i.e. box, meter, or corner joint), (iii) the hand release of the profile, and (iv) the hand moving away. As shown in Figure 6, the IUs were grouped in an analogous way to SCM, no matter who executed the activity. We can deduce that our framework is robust to the variability induced by different subjects in the activity execution. Moreover, the algorithm was robust to changes in the absolute poses of the video objects (see Figure 8). In human-robot interaction scenarios, being able to identify the similarities between human-demonstrated activities can impact the capacity of the robot to understand and eventually replicate the human performance. For instance, similar context-free IUs can be mapped within the same robot motion primitive, reducing the requirements for a large number of pre-defined robot skills.

Additionally, our method shows promising results in leveraging the recognition of both type similar and non-type similar interactions to ensure job performance without anomalies. A preliminary experiment indicated a high success rate and consistent results across iterations using different references (Table I). Interestingly, the errors committed were only false positive, meaning that the algorithm occasionally failed in detecting correctly performed IUs. This was probably due to the limited size of the training set, which may not capture the full range of variations in correct activity executions. With a more extensive training set, we expect to obtain more accurate thresholds that will reduce the rate of false positive and improve the overall performance of the anomaly detection algorithm.

However, we acknowledge several limitations of our approach, including heavy reliance on accurate object and hand detection, the lack of occlusion handling, and the evaluation in 2D with a fixed camera orientation. Moreover, the current version of our method only describes interactions for each hand separately and does not consider any link between activities performed by the two hands, even when they occur simultaneously. In future works, we plan to address these limitations and further refine our method to enhance its capabilities and relevance in robotics applications.

REFERENCES

[1] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, 2011.

[2] J. Massardi, M. Gravel, and É. Beaudry, "Parc: A plan and activity recognition component for assistive robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[3] M. Lagomarsino, M. Lorenzini, P. Balatti, E. D. Momi, and A. Ajoudani, "Pick the right co-worker: Online assessment of cognitive ergonomics in human-robot collaborative assembly," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2022.

[4] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Transferring skills to humanoid robots by extracting semantic representations from observations of human activities," *Artificial Intelligence*, 2017.

[5] P. C. Ribeiro, J. Santos-Victor, and P. Lisboa, "Human activity recognition from video: modeling, feature selection and classification architecture," in *Proceedings of International Workshop on Human Activity Recognition and Modelling*, vol. 61, 2005, p. 78.

[6] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 2015, pp. 1200–1205.

[7] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, 2001.

[8] Y. Fu, A. Ekin, A. M. Tekalp, and R. Mehrotra, "Temporal segmentation of video objects for hierarchical object-based motion description," *IEEE Transactions on Image Processing*, 2002.

[9] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," *Proceedings - International Symposium on Wearable Computers, ISWC*, 2005.

[10] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter, and R. Dillmann, "Action sequence reproduction based on automatic segmentation and Object-Action Complexes," *IEEE-RAS International Conference on Humanoid Robots*, pp. 189–195, 2015.

[11] S. Albrecht, K. Ramírez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," *IEEE-RAS International Conference on Humanoid Robots*, pp. 602–607, 2011.

[12] A. Guha, Y. Yang, C. Fermuuller, and Y. Aloimonos, "Minimalist plans for interpreting manipulation actions," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5908–5914, 2013.

[13] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

[14] A. Fern, J. M. Siskind, and R. Givan, "Learning Temporal, Relational, Force-Dynamic Event Definitions from Video," in *AAAI-02: Eighteenth National Conference on Artificial Intelligence*, 2002.

[15] P. Ardón, È. Pairet, R. P. A. Petrick, S. Ramamoorthy, and K. S. Lohan, "Reasoning on grasp-action affordances," *CoRR*, 2019.

[16] G. e. Carbone, *Grasping in Robotics*, ser. Mechanisms and Machine Science, G. Carbone, Ed. London: Springer London, 2013, vol. 10.

[17] J. Johnson, R. Krishna, M. Stark, L. J. Li, D. A. Shamma, M. S. Bernstein, and F. F. Li, "Image retrieval using scene graphs," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 3668–3678, 10 2015.

[18] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A Comprehensive Survey of Scene Graphs: Generation and Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, 1 2023.

[19] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[20] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, 7 2017.

[21] C. R. Dreher, M. Wächter, and T. Asfour, "Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks," *IEEE Robotics and Automation Letters*, 1 2020.

[22] W. Meert, K. Hendrickx, T. Van Craenendonck, P. Robberechts, H. Blockeel, and J. Davis, "Dtaidistance," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.7158824

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] S. Bhutada, N. Yashwanth, P. Dheeraj, and K. Shekar, "Opening and closing in morphological image processing," *World Journal of Advanced Research and Reviews*, vol. 14, no. 3, pp. 687–695.