2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Categorizing Object-Action Relations from Semantic Scene Graphs

Eren Erdal Aksoy, Alexey Abramov, Florentin Wörgötter, and Babette Dellen

Abstract— **In this work we introduce a novel approach for detecting spatiotemporal object-action relations, leading to both, action recognition and object categorization. Semantic scene graphs are extracted from image sequences and used to find the characteristic main graphs of the action sequence via an exact graph-matching technique, thus providing an event table of the action scene, which allows extracting object-action relations. The method is applied to several artificial and real action scenes containing limited context. The central novelty of this approach is that it is model free and needs *a priori* representation neither for objects nor actions. Essentially actions are recognized without requiring prior object knowledge and objects are categorized solely based on their exhibited role within an action sequence. Thus, this approach is grounded in the affordance principle, which has recently attracted much attention in robotics and provides a way forward for trial and error learning of object-action relations through repeated experimentation. It may therefore be useful for recognition and categorization tasks for example in imitation learning in developmental and cognitive robotics.**

I. INTRODUCTION

One central goal for humanoid robotics is to imitate, understand, and learn from human behavior. Part of this problem is to relate a manipulation to its manipulated object. The difficulty lies here in the fact that individual manipulations even when "doing the same thing" can take vastly different forms just due to changes in posture, action sequence, and/or differences in the general (visual) context surrounding the core manipulation. Nonetheless humans have no problem in classifying manipulation types, such as "moving an object", "closing a book", "making a sandwich" or "filling liquid", and to link objects with actions. The goal of this paper is to devise a method which can, at least to some degree, do the same and thereby classify manipulation types.

Recently, these questions have been approached in an abstract way by the concept of object-action complexes (OACs) [1], [2], claiming that objects and actions are inseparably intertwined. This is linked to the way humans perceive the world by relating objects with actions. The OAC concept proposes such a human-like description by which an object is identified considering both its (visual) properties *and* the actions that have been performed with it. The OAC concept attaches the performed actions to the objects as attributes. This approach is therefore related to the affordance principle [3], which especially in the recent years has had increasing influence in robotics [4]. Take, for example, a cup, which is an entity for filling and drinking. However, not only this single specific cup but any other cylindrical, hollow object could be used for the same actions. Thus, objects, which are supporting common actions, can be considered similar. Filling creates the object-type "container"! Consider now an inverted cup, which cannot be filled. Now the former container has become a pedestal on which you could put something. While physically the same thing, a "pedestal" is a different object type altogether. In cognitive vision, many new approaches for object recognition from 3D models have been introduced [5], [6], [7]. However, these model-based approaches cannot identify object-action relations. In this work, we introduce a novel approach for detecting spatiotemporal object-action relations using semantic scene graphs, leading to both action recognition and object categorization. Using this method, objects are connected to recognized actions considering their roles within a scenario.

The approach relies on a front-end algorithm which allows for the continuous tracking of scene segments using superparamagnetic clustering, with proven convergence properties [8], [9], [10], [11]. The presented core algorithm, used for recognition and classification, then relies on the sequence of neighborhood relations between those segments, which for a given action will always be "essentially" the same. Hence different from feature based (or model based) approaches our system operates on object-part relations without presupposing assumptions about the structure of object and action. Thus, it is model free. This leads to a high degree of invariance against position, orientation, etc. but we need to make sure that segment tracking is stable, which is currently achieved by several means described elsewhere [22]. Furthermore, at this stage we show examples from a 2D (projected) domain. True 3D tracking is currently being implemented for difficult action sequences like "making a complete breakfast".

Therefore, we would like to emphasize that the core contribution of this work is the novel categorization method. The computer vision front end is a required prerequisite, but other tracking methods could be used here as well and improvements are possible.

The structure of the paper is as follows. In Section II we discuss related works. In Section III we introduce the action classification and the object categorization algorithm. In Section IV experimental results with real images are presented.

The work has received support from the BMBF funded BCCN Göttingen, Grant No. 01GQ0430, Project 01GQ0432, and the EU Project PACO-PLUS. We thank Tomas Kulvicius for valuable discussion.

Eren Erdal Aksoy, Alexey Abramov, and Florentin Wörgötter are with Bernstein Center for Computational Neuroscience, University of Göttingen, Friedrich-Hund Platz 1, 37077 Göttingen, Germany eaksoye,abramov,worgott@bccn-goettingen.de

B. Dellen is with Bernstein Center for Computational Neuroscience, Max-Planck Institute for Dynamics and Self-Organization, Bunsenstr. 10, 37073 Göttingen, Germany and Institut de Robotica i Informatica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona, Spain bkdellen@bccn-goettingen.de

In Section V we show directions for future research and how to extend the proposed framework for the process of more complex and longer scenes. Finally, in Section VI the results are discussed.

II. RELATED WORK

Action recognition and object categorization have received increasing interest in the Artificial Intelligence (AI) and cognitive-vision community during the last decade. The problem of action recognition has been addressed in previous works, but only rarely in conjunction with object categorization. Modayil *et al.* (2008) presented a framework focusing on the recognition of activities in daily living [12]. In order to detect the activities, the test subject (e.g. human) was equipped with a Radio Frequency Identification (RFID) reader and tags. The types of actions and used objects were recorded by the RFID reader to learn a model that recognizes the activities performed during observations, and an Interleaved Hidden Markov Model (HMM) was used to increase the accuracy of the learned model. Similar to this study, Liao *et al.* (2005) provided an approach to perform location-based activity recognition by using Relational Markov Networks [13]. This work also covered high-level activities, e.g. working, shopping, dining out, during long periods of time. The system used data from a wearable GPS location sensor and considered time, place of action, and sequence of action, which were extracted from the GPS sensor. Although those kinds of sensor-based multitasking-activity-recognition approaches provide promising results, they do not cover object-categorization issues and have handicaps like limited coverage area. Hongeng (2004) introduced a Markov network to encode the entire event space for scenes with limited context but without considering object classification [14]. Sridhar *et al.* (2008) showed that objects can also be categorized by considering their common roles in actions, resulting however in large and complex activity graphs, which have to be analyzed separately [15]. Li *and* Lee (2000) introduced sub-scene graph matching method just for object recognition, combining it with a Hopfield neural network to get local matches between graphs [16]. Our framework provides a novel approach that represents scenes by semantic graphs which hold spatiotemporal object-action relations. By analyzing semantic scene graphs, we not only recognize actions but also categorize objects based on their action roles.

III. METHODS

A. Overview of the Algorithm

In the current study, we analyze movies of scenes containing limited context. Fig. 1 shows the block diagram of the algorithm. As a first processing step, image segments are extracted and tracked throughout the image sequence, allowing the assignment of temporally-stable labels to the respective image parts [9], [11]. The scene is then described by semantic graphs, in which the nodes and edges represent segments and their neighborhood relations, respectively. For segmentation and graph examples of real images, see Fig. 2. Graphs can change by continuous distortions (lengthening or shortening of edges) or, more importantly, by discontinuous changes (nodes or edges can appear or disappear). Such a discontinuous change represents a natural breaking point: All graphs before are topologically identical and so are those after the breaking point. Hence, we can apply an exact graph-matching method after a breaking point and extract the next following topological main graph. The sequence of these main graphs thus represents all structural changes in the scene. The temporal order by which those main graphs follow each other defines an "event table". An event signifies that something has happened in the scene which caused a true topological change in the graph. This method allows classifying object-action relations by calculating the similarity between event tables from different scenes. Furthermore, nodes playing the same role in an classified action sequence can be identified and then be used to categorize objects by returning to the signal level via image segments.

B. Segmentation and Tracking

We use an image-segmentation method in which segments are obtained trough a 3D linking process [9], [11], [10]. First, a spin variable σ_i is assigned to each pixel i of the stereo image. To incorporate constraints in form of local correspondence information, we distinguish between neighbors within a single frame (2D bonds) and neighbors across frames (3D bonds). We create a 2D bond $(i,k)_{2D}$ between two pixels within the same frame with coordinates (x_i, y_i, z_i) and (x_k, y_k, z_k) if $|(x_i - x_k)| \leq 1$, $|(y_i - y_k)| \leq 1$, and $z_i = z_k$. Across frames, we create a 3D bond $(i,j)_{3D}$ between two spins i and j if $|(x_i + d_{ij}^x - x_j)| \leq 0.5$, $|(y_i + d_{ij}^y - y_j)| \leq 0.5$, $z_i \neq z_j$, and $a_{ij} = 1$. The values d_{ij}^x and d_{ij}^y are the shifts of the pixels between frames z_i and z_j along the axis x and axis y, obtained from an initial optic flow map. The parameters a_{ij} are the respective amplitudes (or confidences). However, since the images in the examples given in this paper are changing only little from frame to frame, we will assume that the flow is zero everywhere. Hence the values d_{ij}^x and d_{ij}^y are zero, and $a_{ij} = 1$ everywhere.

The spin model is now implemented such that neighboring spins with similar color have the tendency to align. We use a q-state Potts model [17] with the Hamiltonian

$$H = -\sum_{\langle ik \rangle_{2D}} J_{ik} \delta_{\sigma_i, \sigma_k} - \sum_{\langle ij \rangle_{3D}} J_{ij} \delta_{\sigma_i, \sigma_j} \quad , \tag{1}$$

with $J_{ij} = 1 - \triangle / \bar{\triangle}$ and $\triangle_{ij} = |g_i - g_j|$, where g_i and g_j are the gray (color) values of the pixels i and j, respectively. The mean distance $\bar{\triangle}$ is obtained by averaging over all bonds.

Here, $\langle ik \rangle_{2D}$ and $\langle ij \rangle_{3D}$ denote that i,k and i,j are connected by bonds $(i,k)_{2D}$ and $(i,j)_{3D}$, respectively. The Kronecker δ function is defined as $\delta_{a,b} = 1$ if $a = b$ and zero otherwise. The segmentation problem is then solved by finding clusters of correlated spins in the low temperature equilibrium states of the Hamiltonian H. The total number M of segments is then determined by counting the computed segments. It is usually different from the total number q

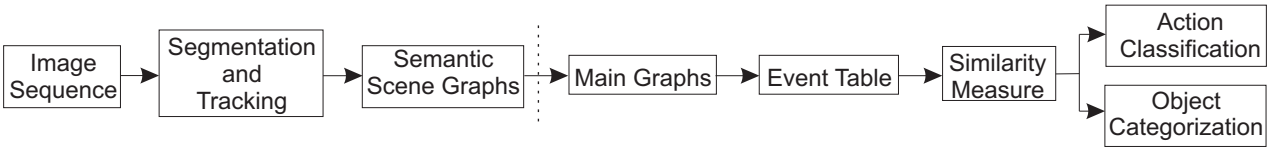


Fig. 1. Block diagram of the algorithm.

of spin states, which is a parameter of the algorithm (here $q = 10$).

We solve this task by implementing a clustering algorithm. In a first step, "satisfied" bonds, i.e. bonds connecting spins of identical spins $\sigma_i = \sigma_j$, are identified. Then, in a second step, the satisfied bonds are "frozen" with a some probability P_{ij}. Pixels connected by frozen bonds define a cluster, which are updated by assigning to all spins inside the same clusters the same new value [18]. In the method of superparamagnetic clustering proposed by [19] this is done independently for each cluster. In this paper, we will employ the method of energy-based cluster updating (ECU), where new values are assigned in consideration of the energy gain calculated for a neighborhood of the regarded cluster [8], [20]. The algorithm is controlled by a single "temperature" parameter, and has been shown to deliver robust results over a large temperature range. After a 100 iterations, clusters are used to define segments.

In this paper, we segment always two consecutive frames of the image sequence at the same time, i.e. frame i and $i+1$, then, we segment the next pair, i.e. $i+1$ and $i+2$, where the last image of the first pair is identical with the first image of the second pair. Then, consecutive pairs are connected by identifying the identical segments in the overlapping images. This strategy allows handling long motion image sequences [11].

C. Semantic Scene Graphs

Once the image sequence has been segmented and segments have been tracked, we represent the scene by undirected and unweighted labeled graphs. The graph nodes are the segment labels and plotted at the center of each segment. The nodes are then connected by an edge if segments touch each other.

Fig. 2 shows original frames with respective segments and semantic scene graphs from four different real action types: *Moving Object*, *Opening Book*, *Making Sandwich*, and *Filling Liquid*. In the *Moving Object* action a hand is putting an orange on a plate while moving the plate together with the orange (see Fig. 2(a-c)). The *Opening Book* action represents a scenario in which a hand is opening a book (see Fig. 2(d-f)). In the *Making Sandwich* action two hands are putting pieces of bread, salami, and cheese on top of each other (see Fig. 2(g-i)). The *Filling Liquid* action represents a scenario in which a cup is being filled with liquid from another cup (see Fig. 2(j-l)).

Larger sample images for each action type are shown in Fig. 3(a-d) to give an impression of the level of complexity, i.e. amount of texture, reflections, and shadows.

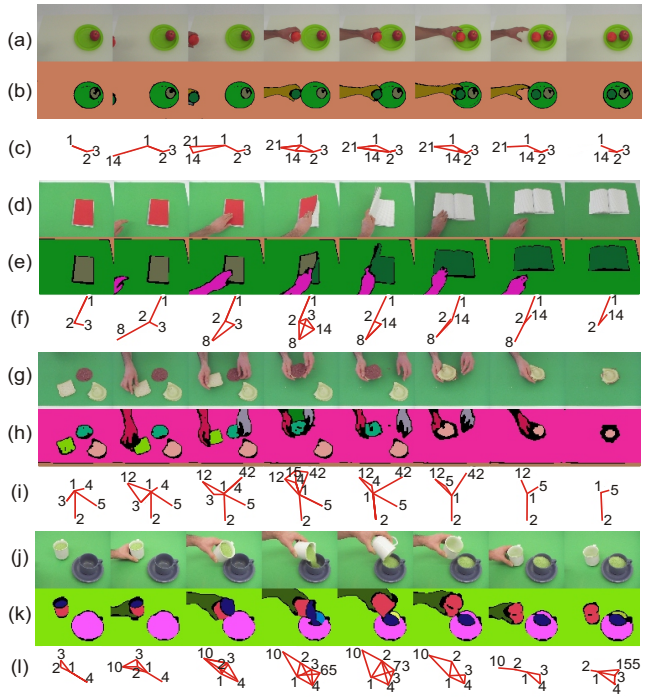


Fig. 2. Four different real action types. (a) Original images from the *Moving Object* action. (b) Respective image segments. (c) Semantic scene graphs. (d) Original images from the *Opening Book* action. (e) Respective image segments. (f) Semantic scene graphs. (g) Original images from the *Making Sandwich* action. (h) Respective image segments. (i) Semantic scene graphs. (j) Original images from the *Filling Liquid* action. (k) Respective image segments. (l) Semantic scene graphs.

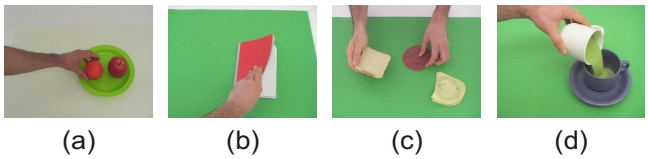
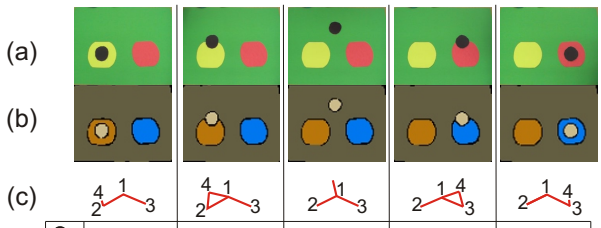


Fig. 3. Sample images taken from each real action type. (a) *Moving Object*. (b) *Opening Book*. (c) *Making Sandwich*. (d) *Filling Liquid*.

D. Main Graphs and Event Tables

In the following we will first use simpler scenes to describe the remaining parts of the algorithm (to the right of the dashed line in Fig. 1). Fig. 4(a-b) depicts original frames with respective segments of an artificial *Moving Object* action (sample action 1) in which a black round object is moving from a yellow vessel into a red vessel.



$\rho_{2,1}$	T	T	O	O	O
$\rho_{3,1}$	O	O	O	T	T
$\rho_{4,1}$	N	T	O	T	N
$\rho_{3,2}$	N	N	N	N	N
$\rho_{4,2}$	O	T	N	N	N
$\rho_{4,3}$	N	N	N	T	O

Fig. 4. Simple example of the *Moving Object* action (sample action 1). (a) Original images. (b) Respective image segments. (c) Semantic scene graphs. (d) Event table.

In the temporal domain, scene graphs represent spatial relations between nodes. Unless spatial relations change, the scene graphs remain topologically the same. The only changes in the graph structures are the node positions or the edge lengths depending on the object trajectory and speed. Consequently, any change in the spatial relation between nodes corresponds to a change in the main structure of the scene graphs. Therefore, those changes in the graphs can be employed to define action primitives. Considering this fact, we apply an exact graph-matching method in order to extract the main graphs by computing the eigenvalues and eigenvectors of the adjacency matrices of the graphs [21]. A change in the eigenvalues or eigenvectors then corresponds to a structural change of the graph. The whole image sequence of the sample *Moving Object* action has 92 frames, however, after extracting the main graphs, only 5 frames are left, each defining a single action primitive (see Fig. 4(c)).

Following the extraction of the main graphs, we analyze the spatial relations between each pair of nodes in the main graphs. We denote the spatial relations by $\rho_{i,j}$ in which i and j are the nodes of interest. Note that the spatial relations are symmetric, i.e. $\rho_{i,j} = \rho_{j,i}$.

Possible spatial relations of each node pair are *absence* (A), *no connection* (N), *overlapping* (O), and *touching* (T). We define those relations by calculating the number of edges of both currently considered nodes i and j in each main graph. As an example, all possible spatial relations between the black object and yellow vessel are illustrated in Fig. 5. Since those objects are represented by graph nodes 4 and 2, we write the relation as $\rho_{4,2}$. The relation *absence* means that one of the considered nodes is not observed in the scene, i.e. the black object node 4 does not exist in the graph (see Fig. 5(a)). In the case of *no connection*, the considered nodes have no edge between them (see Fig. 5(b)). In the *overlapping* relation one of the considered nodes is completely surrounded by the other node. Therefore, the surrounded node has only one edge (see Fig. 5(c)). The *touching* relation represents the situation in which segments

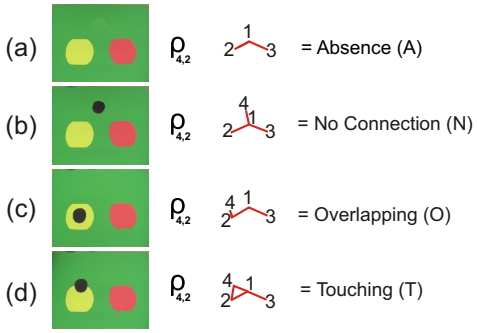


Fig. 5. Possible spatial relations between black object and yellow vessel which are represented by graph nodes 4 and 2, respectively. (a) The relation *absence*. (b) The relation *no connection*. (c) The *overlapping* relation. (d) The *touching* relation.

touch each other and both considered nodes have more than one edge (see Fig. 5(d)). More complex spatial relations between nodes are currently not considered but could be included in the future.

The total number of spatial relations is defined as

$$\rho_{\text{total}} = \sum_{i=1}^{n-1}(n-i) \quad , \tag{2}$$

where n is the total number of objects. For the sample *Moving Object* action mentioned above we have $n = 4$ (yellow and red vessels, a black moving object, and a green background) and therefore $\rho_{\text{total}} = 6$. Those relations are $\rho_{2,1}$, $\rho_{3,1}$, $\rho_{4,1}$, $\rho_{3,2}$, $\rho_{4,2}$, and $\rho_{4,3}$.

All existing spatial node relations in the main graphs are saved in the form of a table where the rows represent spatial relations between each pair of nodes. Since any change in the spatial relations represents an event that defines an action, we refer to this table as an *event table* (ξ). Fig. 4(d) shows the *event table* of the action above. However, the fourth row of the *event table* does not hold any change in the sense of a changing spatial relation since the yellow and red vessels never move. For this reason, we ignore the fourth row. For the sake of simplicity, we substitute numbers -1, 0, 1, and 2 for possible spatial relations A, N, O, and T. The final *event table* of sample action 1 is given in Table 1.

E. Similarity Measure

So far we showed how to represent a long image sequence by an event table the dimensions of which are related to the spatial node relations in the main graphs. Next we will

$\rho_{2,1}$	2	2	1	1	1
$\rho_{3,1}$	1	1	1	2	2
$\rho_{4,1}$	0	2	1	2	0
$\rho_{4,2}$	1	2	0	0	0
$\rho_{4,3}$	0	0	0	2	1

TABLE I

EVENT TABLE (ξ_1) OF THE FIRST SAMPLE ACTION. SPATIAL RELATIONS BETWEEN THE NODES OF SAMPLE ACTION 1.

discuss how to calculate the similarity of two actions. To this end we created one more sample for the *Moving Object* action. Fig. 6 depicts the main graphs of sample action 2 in which a red rectangular object is moving from a blue vessel into a yellow vessel following a different trajectory with different speed as compared to the first sample. Moreover, the scene contains two more objects which are either stationary (red round object) or moving randomly (black round object). Following the same procedure, the *event table* for the second sample is calculated and given in Table 2. Note that even though the second sample contains more objects, the dimension of the event tables is accidentally the same. This makes explanations simpler, but, as we will see later, the dimensions of the event tables are not important and can even be different between two cases.

Similarity measurement of actions is based on the comparison of the event tables. Basically, each row of the first *event table* (ξ_1) is compared with each row of the second *event table* (ξ_2) in order to find the highest similarity. (For event tables with different dimensions, sub-matrices need to be used.) Considering this simple rule we start determining the similarity with the first rows of ξ_1 and ξ_2, giving [2 2 1 1 1] and [1 1 1 2 2], respectively. Those lines are written one below the other. Next, the amount of equal digits (equal relations!) are counted and divided by total number of digits. Since only one digit (third digit) out of five digits is the same, the similarity of those two rows is 20%. Once all rows have been compared with each other, the determined similarity values are saved in the form of a table where rows and columns give the similarity relations between ξ_1 and ξ_2. The resulting table is called *similarity table* (ζ) and shown in Table 3.

The final similarity measure is determined by calculating the arithmetic mean value of the highest values in each

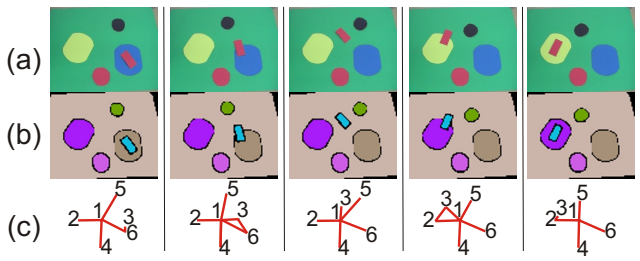


Fig. 6. Different version of the simple *Moving Object* action (sample action 2). (a) Original images. (b) Respective image segments. (c) Semantic scene graphs.

$\rho_{2,1}$	1	1	1	2	2
$\rho_{3,1}$	0	2	1	2	0
$\rho_{6,1}$	2	2	1	1	1
$\rho_{3,2}$	0	0	0	2	1
$\rho_{3,6}$	1	2	0	0	0

TABLE II

EVENT TABLE (ξ_2) OF THE SECOND SAMPLE ACTION. SPATIAL RELATIONS BETWEEN THE NODES OF SAMPLE ACTION 2.

ξ_1 \ ξ_2	$\rho_{2,1}$	$\rho_{3,1}$	$\rho_{6,1}$	$\rho_{3,2}$	$\rho_{3,6}$
$\rho_{2,1}$	20%	40%	**100%**	20%	20%
$\rho_{3,1}$	**100%**	40%	20%	20%	0%
$\rho_{4,1}$	40%	**100%**	40%	40%	40%
$\rho_{4,2}$	20%	40%	20%	20%	**100%**
$\rho_{4,3}$	20%	40%	20%	**100%**	20%

TABLE III

SIMILARITY TABLE (ζ). SIMILARITY VALUES BETWEEN ξ_1 AND ξ_2.

row of ζ. Consequently, our two sample actions have 100% similarity.

In case of having event tables with different dimensions, we apply a window-based search algorithm to the bigger table in order to find out a region that has the highest similarity with the smaller table. In this case, the number of total search is defined as

$$s_{\text{total}} = (|r_1 - r_2| + 1)(|c_1 - c_2| + 1) \quad , \tag{3}$$

where r_1, r_2, c_1, and c_2 are the row and column numbers of the first and second event tables. The final similarity measurement is the highest similarity observed during this total search. If the dimensions are inconsistent in size to decide which one is smaller (such as $r_1 < r_2$ and $c_1 > c_2$ or $r_1 > r_2$ and $c_1 < c_2$), the event table with less columns is extended by adding the last column until it has the same number of columns as the bigger table. This sort of operation does not affect the action content since we do not change spatial node relations in the temporal domain.

As a result we can now measure how similar the two actions are and we find 100%. Thus, these actions are of the same type ("type-similar").

F. Object Categorization

The *similarity table* also implicitly encodes the similarity of the nodes between the two different examples. Intriguingly, this can be used to extract *nodes with the same action roles* in type-similar actions. For this we first list all relations ρ of both actions with highest individual similarity. For instance, the relation between nodes 2 and 1 ($\rho_{2,1}$) in the first row has a 100% similarity with the relation between nodes 6 and 1 ($\rho_{6,1}$) in the third column. Doing this for all relations, we find the following maximal similarities in ζ:

$$\begin{aligned}
\rho_{2,1} &\Leftarrow 100\% \Rightarrow \rho_{6,1} \\
\rho_{3,1} &\Leftarrow 100\% \Rightarrow \rho_{2,1} \\
\rho_{4,1} &\Leftarrow 100\% \Rightarrow \rho_{3,1} \\
\rho_{4,2} &\Leftarrow 100\% \Rightarrow \rho_{3,6} \\
\rho_{4,3} &\Leftarrow 100\% \Rightarrow \rho_{3,2} \quad .
\end{aligned}$$

Those similarity values represent the correspondences between manipulated nodes in ξ_1 and ξ_2. In order to determine these correspondences, we analyze which node number in ξ_1 is repeating in conjunction with which node number in ξ_2. We start with node number 1 in ξ_1, and obtain

$$\begin{array}{lll} \rho_{2,1} & \Leftarrow 100\% \Rightarrow & \rho_{6,1} \\ \rho_{3,1} & \Leftarrow 100\% \Rightarrow & \rho_{2,1} \\ \rho_{4,1} & \Leftarrow 100\% \Rightarrow & \rho_{3,1} \end{array} \quad \Rightarrow \quad 1 \approx 1 \quad .$$

While 1 is repeating three times in ξ_1, the same node number 1 in ξ_2 is also repeating three times. However, node numbers 2, 3, and 6 in ξ_2 occur only once. Therefore, we conclude that graph nodes 1 in both ξ_1 and ξ_2 had the same roles. In fact, both graph nodes represent the green background which plays same role in both actions.

We continue the spatial node relation analysis with node number 2 in ξ_1, and obtain

$$\begin{array}{lll} \rho_{2,1} & \Leftarrow 100\% \Rightarrow & \rho_{6,1} \\ \rho_{4,2} & \Leftarrow 100\% \Rightarrow & \rho_{3,6} \end{array} \quad \Rightarrow \quad 2 \approx 6 \quad .$$

Node number 2 in ξ_1 is repeating twice with node number 6 in ξ_2. Those graph nodes represent the yellow and blue vessels within which the moving objects are initially located and from which they then move away.

For the case of node number 3 in ξ_1 we obtain

$$\begin{array}{lll} \rho_{3,1} & \Leftarrow 100\% \Rightarrow & \rho_{2,1} \\ \rho_{4,3} & \Leftarrow 100\% \Rightarrow & \rho_{3,2} \end{array} \quad \Rightarrow \quad 3 \approx 2 \quad .$$

Node number 3 in ξ_1 corresponds to node number 2 in ξ_2 because both of them are repeating twice. Those graph nodes define the destination vessels for the moving objects.

The last node number 4 in ξ_1 is obtained as

$$\begin{array}{lll} \rho_{4,1} & \Leftarrow 100\% \Rightarrow & \rho_{3,1} \\ \rho_{4,2} & \Leftarrow 100\% \Rightarrow & \rho_{3,6} \\ \rho_{4,3} & \Leftarrow 100\% \Rightarrow & \rho_{3,2} \end{array} \quad \Rightarrow \quad 4 \approx 3 \quad .$$

As node number 4 in ξ_1, node number 3 in ξ_2 is also repeating three times. In fact, both graph nodes represent the moving objects which are the round black object in ξ_1 and the rectangular red object in ξ_2.

In the case of having a *similarity table* which has the same highest value more than once in a column, e.g. having two times 100% similarity values in the same column, the object categorization section leads to ambiguous results, i.e. one object corresponds to two different objects. Since this sort of correspondence is not allowed in the framework, the final similarity value is calculated again by taking the second highest values into account. This way we can get rid of any kind of mismatching in the object categorization process.

IV. Results with Real Images

We applied our framework to four different real action types: *Moving Object*, *Opening Book*, *Making Sandwich*, and *Filling Liquid* (see Fig. 2). For each of these actions, we recorded four movies with different trajectories, speeds, hand positions, and object shapes. All those sixteen movies were recorded by a stable camera that was focused on the hands and the manipulated objects.

In Fig. 7, some sample frames of all four action types are shown which are different from those in Fig. 2. Here, for

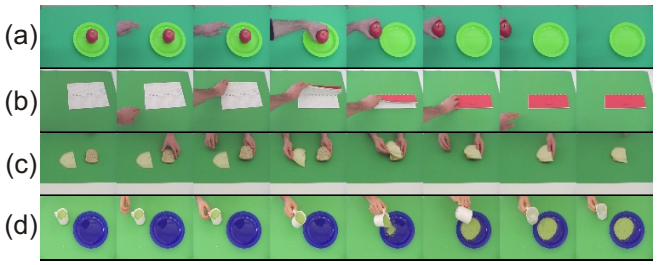


Fig. 7. Different versions of the real action types. (a) *Moving Object*. (b) *Opening Book*. (c) *Making Sandwich*. (d) *Filling Liquid*.

the *Moving Object* action a hand is *appearing* in the scene, taking an apple from a plate, and leaving the scene (see Fig. 7(a)). The *Opening Book* action type represents here a scenario in which a hand is *closing* a book (see Fig. 7(b)). In the *Making Sandwich* action two hands are putting pieces of bread and cheese on top of each other in *different* order (see Fig. 7(c)). The *Filling Liquid* action type includes a scenario in which a cup is filling a *plate* with liquid (see Fig. 7(d)). These examples were introduced to show that really different instantiations of a manipulation will still be recognized as belonging to the same type.

Event tables of each real test data are compared with each other. The resulting similarity values are given in Fig. 8. Each test data has at least 69% similarity with the other versions of its type-similar action (see close to diagonal). In general the similarity between type-similar actions is for all scenes much bigger than the similarity between non-type-similar actions, except in one case. For the fourth version of *Making Sandwich* and the fourth version of *Moving Object* we receive a large similarity of 57%. This may happen in some cases when action primitives are quite similar and, in addition, noise in the data leads to a few spurious nodes and

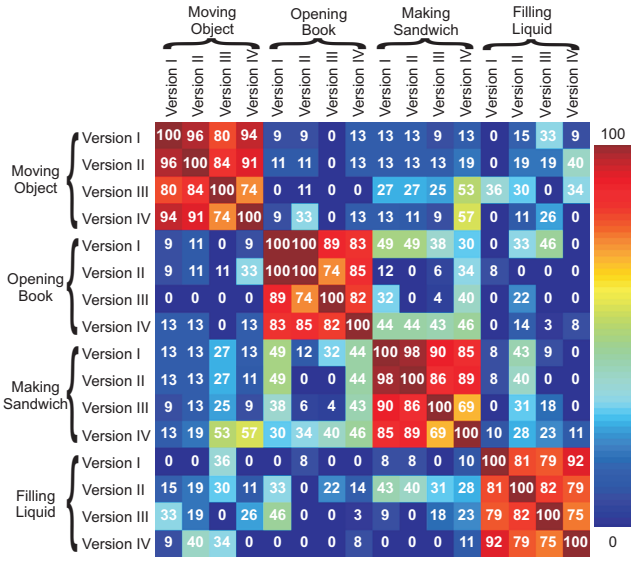


Fig. 8. Action-classification results. Similarity values between event tables of the real test data set.

false tracking. During tests with real images the accuracy of the whole algorithm is observed as 91%.

Moreover, the results showed that the manipulated objects in each action type can be categorized according to their roles in the actions. Fig. 9 illustrates the categorization results of objects that performed same roles in different versions of actions. As an example, the apple and orange are in the same group since they are being manipulated in the *Moving Object* action.

Notwithstanding some remaining problems, the results shown here clearly demonstrate that it is possible to classify objects and actions in scenes with limited context without prior (model) knowledge.

V. Extending the Algorithm

So far we showed how the algorithm works in 2D. Next we will discuss how to extend the algorithm to 3D in order to apply our framework to more complex and even longer scenes containing high-level context. To this end we have to introduce model free stereo-video segmentation and stereo segment tracking leading to 3D scene graphs [22]. As an example, Fig. 10(a) shows sample frames of disentangling a complete "making a breakfast" stereo sequence. In this scenario two arms are first taking a bread from a toaster, putting a piece of cheese on it, and then cutting off a slice of salami with a knife. After putting the salami on top of the cheese, the sandwich is being placed on a plate and the arms are leaving the scene. Respective image segments of the sample frames from the left stereo image sequences are given in Fig. 10(b). The corresponding dense disparity maps obtained for extracted stereo segments are shown in Fig. 10(c) [10]. Therein the low-confidence-value area of the table segment is depicted with a black color. Fig. 10(d) illustrates 3D semantic scene graphs of the selected frames. In 3D graphs, edges show that the segments are neighbors and their depth differences are less than a predefined threshold value. While the number of segments might much increase in more complex scenes, the number of *consistently changing* edges will remain small as real 3D changes of touching relations (valid derivatives) are rare. Thus we are currently improving the presented algorithm in two ways: 1) We are decomposing action sequences into action sub-sequences and analyze each of them separately and 2) We are also compressing the event tables by taking their derivatives. This way we are decreasing complexity and also computation time.

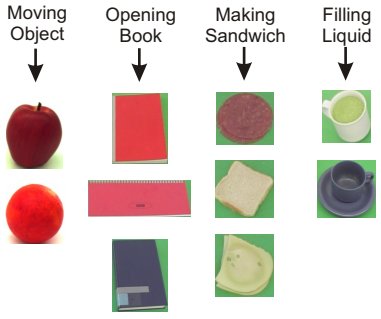


Fig. 9. Object categorization results. In each action type the manipulated objects can be detected based on their action roles.

Furthermore, determining similarity values between each action makes the whole system computational expensive, especially if the database contains a lot of training data. In order to avoid this problem, we plan to construct a template main-graph model for each kind of action. Template graph models can be constructed by considering the main graphs of a scenario which accurately represents the respective action type. Actions will then be classified by calculating the similarity values with those template models instead of with one another. In addition to this, we intend to let the agent learn the template main-graph models from a training data set. To achieve all this a parallel implementation of the framework on GPUs for real-time robotics applications is currently being implemented.

VI. Discussion

We presented a novel algorithm that represents a promising approach for recognizing actions *without* requiring prior object knowledge, and for categorizing objects solely based on their exhibited role within an action sequence. Our framework is mainly based on the analysis of object relations in the spatiotemporal domain. We are aware of the fact that "segment permanence" (i.e., reliable tracking) needs to be assured without which our method would fail. Clearly on the computer vision side improvements can be made to better assure this. This, however, is not the point of this paper. As far as we see it this is one of the first papers in which the categorization of object-action relations has become possible in a model free way. This procedure can thus be entirely based on the experimentation of the robot (here simulated by a human). Hence we arrive at a very high sub-symbolic representational level in a fully grounded way. From there on the grounded development and the learning of symbols (for example verbal utterances) which describe actions should be easier than before and this has been deemed as one of the major challenges in cognitive robotics. Furthermore, it should also be possible to "backwards unwrap" the learned event tables (the OACs) and this way *generate* an action. Obviously complex inverse kinematic (and dynamic) problems need to addressed to arrive at an actual movement sequence. However, the event graphs specify the fundamental "breaking points" whenever certain object relations change. Therefore movement segments between two such breaking points could be seen as motor primitives. The execution of such a primitive may then be optimized by whatever means but one always must assure that its starting point (prior) and its endpoint (posterior) corresponds to two subsequent entries in the event table.

The proposed algorithm has been applied to four different real action sequences of scenes containing limited context. Each action type had four different versions which differed in trajectories, speeds, hand positions, and object shapes. The experimental results showed that the agent can categorize all these action types by measuring the amount of similarity between action sequences and also categorize the participating

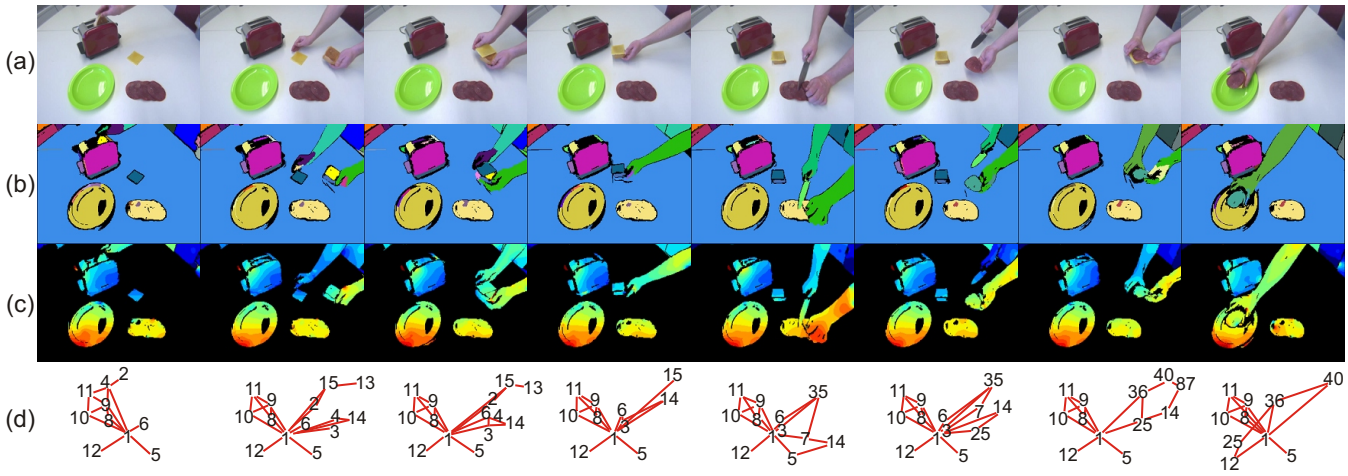


Fig. 10. A sample "making a breakfast" stereo sequence. (a) Original frames from the left image sequence. (b) Extracted segments for frames of the left sequence. (c) The dense disparity maps obtained for extracted stereo segments. The disparity values are color-coded from blue (small) to red (large). Areas of low confidence are colored black, i.e., the uniform and untextured area of the table, for which only poor disparity results could be obtained. (d) Final 3D semantic scene graphs.

manipulated objects according to their roles in the actions.

Several extensions of this algorithmic framework will be pursued in the future as we discussed above.

In summary, this study is one of the first to show that it is indeed possible to treat objects and actions as conjoint entities as suggested by the abstract idea of object-action complexes (OACs, [1], [2]). This is the first description of our new approach and the discussion above shows that it seems to have high potential. In general this contribution shows that this complex concept is algorithmically treatable and therefore we believe that the OAC indeed provides a promising approach for treating problems involving cause-effect relations in cognitive robotics.

REFERENCES

[1] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive agents - a procedural perspective relying on predictability of object-action complexes (oacs)," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 420–432, 2009.

[2] N. Krüger, J. Piater, C. Geib, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrcen, A. Agostini, and R. Dillmann, "Object-action complexes: Grounded abstractions of sensorimotor processes (submitted)," *Robotics and Autonomous Systems*, 2009.

[3] J. Gibson, "The ecological approach to visual perception." Boston: Houghton Mifflin, 1979.

[4] S. Hart and R. Grupen, "Intrinsically motivated affordance learning," in *Workshop on Approaches to Sensorimotor Learning on Humanoids, IEEE Conference on Robotics and Automation (ICRA)*, 2009.

[5] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2004.

[6] C. H. Andez, C. Hern, E. Esteban, and F. Schmitt, "Silhouette and stereo fusion for 3d object modeling," *Computer Vision and Image Understanding*, vol. 96, pp. 367–392, 2004.

[7] M. Tomono, "3d object modeling and segmentation based on edge-point matching with local descriptors," in *ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 55–64.

[8] R. Opara and F. Wörgötter, "A fast and robust cluster update algorithm for image segmentation in spin-lattice models without annealing - visual latencies revisited," *Neural Computation*, vol. 10, pp. 1547–1566, 1998.

[9] N. Shylo, F. Wörgötter, and B. Dellen, "Ascertaining relevant changes in visual data by interfacing ai reasoning and low-level visual information via temporally stable image segments," in *Proceedings of the International Conference on Cognitive Systems (Cogsys 2008)*, 2009.

[10] B. Dellen and F. Wörgötter, "Disparity from stereo-segment silhouettes of weakly textured images," in *Proceedings of the British Machine Vision Conference*, 2009.

[11] B. Dellen, E. E. Aksoy, and F. Wörgötter, "Segment tracking via a spatiotemporal linking process in an n-d lattice model," *Sensors*, vol. 9, no. 11, pp. 9355–9379, 2009.

[12] J. Modayil, T. Bai, and H. Kautz, "Improving the recognition of interleaved activities," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 40–43.

[13] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition using relational markov networks," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 773–778.

[14] S. Hongeng, "Unsupervised learning of multi-object event classes," in *Proc. 15th British Machine Vision Conference*, 2004, pp. 487–496.

[15] M. Sridhar, G. A. Cohn, and D. Hogg, "Learning functional object-categories from a relational spatio-temporal representation," in *Proc. 18th European Conference on Artificial Intelligence*, 2008.

[16] L. Wen-Jing and L. Tong, "Object recognition by sub-scene graph matching," in *ICRA '00: Proceedings of the 20th IEEE International Conference on Robotics and Automation*, 2000.

[17] R. B. Potts, "Some generalized order-disorder transformations," *Proc. Cambridge Philos. Soc.*, vol. 48, pp. 106–109, 1952.

[18] R. H. Swendsen and S. Wang, "Nonuniversal critical dynammics in monte carlo simulations," *Physical Review Letters*, vol. 76, no. 18, pp. 86–88, 1987.

[19] M. Blatt, S. Wiseman, and E. Domany, "Superparametric clustering of data," *Physical Review Letters*, vol. 76, no. 18, pp. 3251–3254, 1996.

[20] C. von Ferber and F. Wörgötter, "Cluster update algorithm and recognition," *Physical Review E*, vol. 62, pp. 1461–1664, 2000.

[21] M. F. Sumsi, "Theory and algorithms on the median graph. application to graph-based classification and clustering," Ph.D. dissertation, Universitat Autonoma de Barcelona, 2008.

[22] A. Abramov, E. E. Aksoy, B. Dellen, J. Doerr, and F. Wörgötter, "3d semantic representation of actions from efficient stereo-image-sequence segmentation on gpus (submitted)," in *3DPVT*, 2010.