A New Bayesian Modeling for 3D Human-Object Action Recognition

Camille Maurice, Jorge Francisco Madrigal Diaz, André Monin, Frédéric Lerasle

A new Bayesian modeling for 3D human-object action recognition

Camille Maurice, Francisco Madrigal, André Monin
LAAS-CNRS
7 avenue du Colonel Roche, 31400, Toulouse, France
name.surname@laas.fr

Frédéric Lerasle
University Paul Sabatier
Route de Narbonne, 31330, Toulouse, France
lerasle@laas.fr

Abstract

Intelligent surveillance systems in human-centered environments require people behavioral monitoring. In this paper, we propose a new Bayesian framework to recognize actions on RGB-D videos by two different observations: the human pose and objects in its vicinity. We design a model for each action that integrates these observations and a probabilistic sequencing of actions performed during activities. We validate our approach on two public video datasets: CAD-120 and Watch-n-Patch. We show a performance gain of 4% in action detection on the fly on CAD-120 videos. Our approach is competitive to 2D image features and skeleton-based methods, as we present an improvement of 16% on Watch-n-Patch. Action recognition performance is clearly improved by our Bayesian and joint human-object perception.

1. Introduction

Activity recognition is an important task in the development of many practical applications such as home health monitoring, human-robot interaction, among others. An activity can be seen as a temporal sequence of actions *e.g.*: the activity *prepare coffee* involves actions such as *pour* water, *add* ground coffee and *turn on* machine. Activities involved at home or in an industrial environment may differ but their underlying actions may be similar as they cover object displacement, object grasping, object interaction...

This is still a challenging topic where actions could be misinterpreted due to background clutter, partial occlusions of the body, presence of numerous objects in the human vicinity and viewpoint changes. Some of the problems can be handled by the use of additional sources, i.e., wearable devices such as accelerometers [3], but the possibility to use invasive equipment depends on the context or sometimes is not applicable.

Depth cue offers non-invasive information with the opportunity to segment the background and to create 3D models invariant to camera viewpoints. From these insights, we focus on non-physically invasive sensors and decide to only rely on a single low-cost RGB-D sensor (*e.g.* Kinect, XTion).

Convolutional Neural Network (CNN) based methods have shown good performance in the area of multiple objects detection and human pose estimation over color images. The fusion of both objects and human pose perception can be used to understand complex activities, especially in the case when objects are in interaction with the environment and the human pose trajectory is not highly discriminant alone. Thus, building upon the recent advances of CNNs we propose to use human pose and objects detection as low-level input for our Bayesian framework which allows labelling actions on the fly.

Some actions do not differ much in the human pose involved but rather in the object configuration during action execution. Given the inferred class of each object, we can provide knowledge on how it is manipulated *i.e.* the so-called affordance [5]. Multiple affordance labels can be associated with an object class *e.g.* an apple is *eatable* and *peelable* not *readable*. Those affordance labels may imply specific object configuration, in the case of the label *peelable* there must be a peeler or a knife nearby.

However human pose perception is relevant to achieve action recognition as some actions can be discriminated only by the pose. Grasping action type has a typical movement of the hand going away from the body to reach an object. Using the pose information, especially when all objects are static we can find which one is in interaction. Also,

human pose information can help to reduce erroneous object movement due to flickering values in the depth image.

Our contributions in this work involve (1) the integration of contextual cues through human pose and object detection by open-source deep learning libraries into a Bayesian framework for action detection.(2) A model of actions reusable into a different context.(3) The use of a hyperparameter optimization tool for tuning.

The rest of the paper is organized as follows. We briefly review the related work in Section 2. We then describe our joint human-object Bayesian modeling and details its implementation in Section 3. We present our evaluation protocol and highlight our gains over two public datasets in Section 4. We draw a conclusion of the presented work in Section 5.

2. Related Work

Early approaches [30, 15] and datasets [14, 15] focus solely on human perception and its pose trajectories in video sequences for action classification. Each video in those datasets contains a single human action without any context: no backgrounds nor surrounding objects. Still today, most of the largest RGB-D datasets focus on skeleton-based action detection with few to no objects to interact with, *e.g.*: MSRAction [15], BerkeleyMHAD [21], Human3.6M [8], NTU RGB+D [25]. Even some datasets offer more realistic scenarios where a single person performs, in the same video sequence, different actions that may involve interaction with his/her environment, *e.g.* object manipulation, object-to-object interaction. This kind of scenario with contextual information can be found in CAD-120 [13] and Watch-n-Patch [28] datasets.

Human action recognition approaches can be roughly categorized into data-driven or model-driven paradigms. Data-driven approaches such as Convolutional Neural Networks (CNN) for action recognition [1, 10, 27], pose and object detection [2, 18] are state-of-the-art on 2D videos or single image datasets. These deep learning networks learn a representation of a model based on pre-recorded and labeled data, the so-called supervised learning.

Recent advances in 2D deep learning object (resp. human pose) detection with frameworks such as Single Shot Detector [18] (resp. OpenPose [2]) are also linked to the emergence of challenges. An example is the MSCOCO challenge [16] which provides images with 1.7 million of labeled keypoints for human pose detection and also images with 500k annotated objects. Proposals have demonstrated their robustness to a variety of situations and camera viewpoints by means of a heavy training over large annotated datasets.

Since the release of NTU-RGB+D [25], the largest RGB-D dataset for human action recognition, most of the 3D deep learning approaches are solely based on 3D human pose through the analysis of skeleton trajectories [17]. However, those techniques are not applicable on contextual action recognition for two reasons: (1) Sometimes skeleton motion alone cannot discriminate between two different actions, i.e., *eating* an apple and *drinking* water imply similar motion of the hand towards the head. (2) They are not straightly applicable due to the limited size of 3D datasets such as CAD-120 and Watch-n-Patch.

To handle the limited size of the aforementioned datasets, H. Kataoka *et al*. [11] employ color and differential images as input of a CNN pre-trained on ImageNet (over 1 million annotations) to perform action detection on Watch-n-Patch. M. S. Ryoo *et al*. [24] combines 2D CNN features pre-trained on ImageNet and conventional motion features such as Histograms of Optical Flow (HOF). Even though the dataset provides depth information, both approaches [11] [24] rely only on 2D image and motion features. A. Jain *et al*. [9] use a Structural Recurrent Neural Network (S-RNN) with pre-defined spatio-temporal graph structures where nodes and edges are represented as Long Short-Term Memory units (LSTM). They reason on 3D skeletons and their interaction with objects.

While achieving undisputed performances when training data is sufficiently diverse and large, end-to-end deep-learning action recognition frameworks suffer from a lack of interpretability of their features [26]. When a CNN does not recognize an action, it is difficult to know whether the issue is related to the implicit inference of the object detection and/or related to human pose. Their dependency on a large volume of training data implies high computational costs and a lack of explanatory capacities.

For context-based action recognition, we are interested in the spatial configuration of the objects and their interaction with other objects of the scene and with the human. RGB-D videos propose a 3D segmentation of the scene based on the involved distances (sensor vs. scene, human vs. objects, object vs. object). The 3D perception of the scene allows us to derive distances in real units (*e.g.* meters) and is less subject to misinterpretation of the scene due to perspective effects. Thereby, irrelevant objects in the background that may lead to a mistake in the action recognition are easily removed. Furthermore, action motions are more discriminant with 3D information especially when the motion is not fronto-parallel.

Whereas model-based algorithms are inherently interpretable, they also require less training data. Furthermore, model-based approaches may include custom modeling of skeleton trajectories, affordances and spatial configuration of the scene. H. Koppula and A. Saxena [12] propose a Conditional Random Field (CRF) that models the scene and the spatial-temporal relations through object affordances.

To recognize actions over contextual RGB-D datasets, we design Bayesian models to infer the action labels on

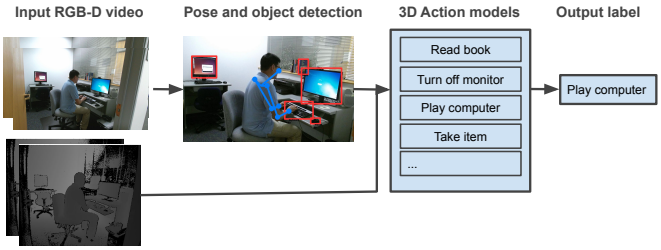


Figure 1. Action detection framework.

video streams. The scene reveals different cues about the nature of objects, their spatial configuration in the human vicinity, and his/her posture. The previous action also gives information about the likelihood of the following actions. Our framework considers all the aforementioned cues. We take the opportunity of the recent development of 2D deep-learning for human pose and object detection to use them as lower input for a higher-level inference: the current action on the video. No further data labeling or training is required as pre-trained deep-learning models are widely available. Our framework is tuned from a small set of annotated data, using a Bayesian optimization tool: SMAC [7]. Thus our approach presents adaptability characteristics that ease the addition of new action classes or the use of another dataset. To the best of our knowledge, such a framework *i.e*. sequencing deep-learning (data driven) and Bayesian (model driven) concepts has not been designed before in the literature.

Fig. 1 illustrates our action recognition framework and we present the probabilistic action models in the following section.

3. Our Approach

In this section, we first define the probabilistic model, which is the core of our action detection process, then we provide details regarding its implementation.

3.1. Probabilistic Formulation

Here we consider activities performed by a single human with several objects known in a priori. An activity is defined by a sequence of actions as proposed by Moeslund*et al*. [20]. We recognize the actions based on the joint observation of the human pose and the objects in the scene. We describe the human pose by a skeleton composed by a set of 3D points representing joints and some pairs of joints represent limbs.

We consider there are two types of actions to infer: (1) actions before a hand grasps an object and (2) actions while a hand is in contact with an object. In both cases, the whole surrounding objects and skeleton pose affect the probabilities of action classes. Object affordances are considered in the human-object interactions as well as in object-object interactions. Once an object is grasped it can further interact with: a part of the body (*e.g*. to eat fruit with the mouth, to make a phone call with the ear) or with another object (*e.g*. pour a bottle of milk into a cup, place a book on the table). No further interaction is also an alternative, (*e.g*. move an object, use a smartphone, turn on a light.) Our approach is designed to encompass all these different types of actions in a unified Bayesian framework based on the inferred observations of the spatial configuration of the objects and the human pose.

Each action a is associated with a model. Let $\mathrm{A} = \{a^1, a^2, ..., a^n\}$ be the set of n actions. Let $O_t = \{s_t, \Omega_t\}$ represent, at time t, the joint observation of the human skeleton s_t and the set of objects detected in the human vicinity: $\Omega_t = \{\omega^1, \omega^2, ..., \omega^{Card(\Omega)}\}$ with $Card(\Omega)$ being the number of objects in the scene. We model the *a posteriori* probability of the actions given the observations as follows:

$$p(a_{0:T}|O_{0:T}) \propto \prod_{t=0}^{T} p(O_t|a_t) \prod_{t=1}^{T} p(a_t|a_{t-1}). \quad (1)$$

In equation (1), $p(O_t|a_t)$ is the likelihood of the observation given the action a_t. The term $p(a_t|a_{t-1})$ denotes the transition probability between two successive actions. Thus, our approach aims to achieve online action classification, without the need of an action segmentation preprocessing step. The classification is inferred through information aggregation over T frames. Therefore, the inference is near to real-time as T typically ranges between 2 and 10 frames.The observation likelihood in equation (1) is decomposed as:

$$p(O_t|a_t) \propto p(s_t|a_t, \Omega_t) p(\Omega_t|a_t). \quad (2)$$

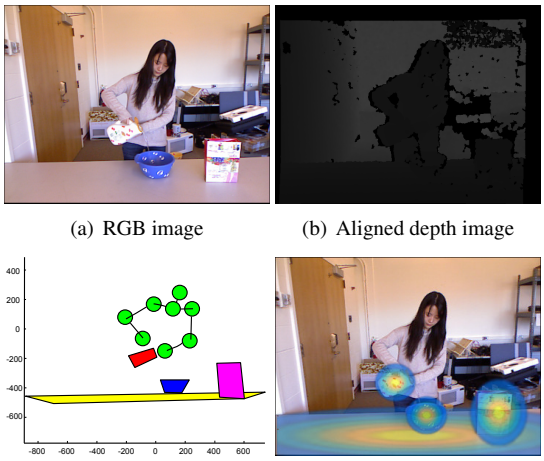
Here in equation (2), $p(s_t|a_t, \Omega_t)$ models the probability of the observed skeleton (s_t) to be a representation of the estimated skeleton model associated with action a_t, in interaction with one object of the set Ω. The skeleton pose s is in our case is reduced to the 3D positions of the upper body extremities *i.e*. hands and head. This is represented as a 9-dimensional vector of Cartesian joint positions in the world coordinate frame as shown in equation (3):

$$s = (s_{\text{l.hand}},\ s_{\text{head}},\ s_{\text{r.hand}}). \quad (3)$$

The equation (2) is the product of two likelihoods, the first one can be expressed as:

$$p(s_t|a_t, \Omega_t) \propto \mathcal{N}(d_p(s, s_n); \mu_1, \sigma_1) \mathcal{N}(d_o(s, \Omega); \mu_2, \sigma_2), \quad (4)$$

where s_n is the skeleton model associated with action a_t, for example if a_t = eat, one of the hand should be close to the head. For clarity reasons we omit the term t. In



(a) RGB image (b) Aligned depth image

(c) 3D Spatial configuration I (d) Spatial configuration II

Figure 2. From the Kinect camera RGB-D input of 2(a) and 2(b) we model the 3D object configuration of the scene as shown on Figs. 2(c) and their associated Gaussian distribution as shown on 2(d).

this equation, d_p is the 3D pose distance between the skeleton model and the estimated skeleton pose and d_o is the 3D distance between the skeleton hands and the objects. This second normal distribution represents the probability of an object in hand to be in interaction with other objects in the scene. Standard deviations σ_1, σ_2 on this equation are hyperparameters that need to be tuned for each action. Object affordances and object-object interactions are modeled through the position of the hands.

The second likelihood of equation (2), *i.e.* $p(\Omega_t|a_t)$, represents the spatial configuration of all objects detected on the scene and it is modeled as follows:

$$p(\Omega_t|a_t) = \prod_{k=1}^{\mathrm{Card}(\Omega_t)} p(\omega_t^k|a_t). \tag{5}$$

In equation (5), the likelihood is also expressed as a normal distribution.

In Fig. 2 we present the RGB-D frames input, the 3D spatial configuration of objects as well as associated Gaussian probability distributions. We can see during the action *pour* that static objects are in interaction with the table, and the milk is in interaction with the bowl while being in the hands. Details regarding human pose and object detection, as well as the parameter optimization of our modeling, are in the following section.

3.2. Implementation

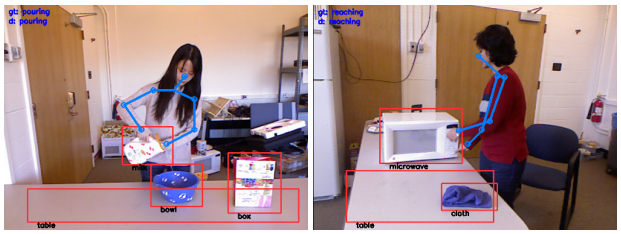
As mentioned in Section 3.1 our formulation requires the skeleton s as a representation of the human pose and the objects ω observed. They are both detected using 2D images through two deep learning neural network frameworks and projected onto the 3D space using the given calibration parameters of the Kinect camera. The 2D human pose is inferred by OpenPose [2], a popular open source library with up to 900 citations, that outperforms state-of-the-art results while maintaining real-time execution. The authors also released pre-trained models on MSCOCO [16] keypoints challenge data. This framework takes as input an RGB image to infer the human posture, *i.e.* the localization in the 2D image of 25 skeleton joints and their corresponding confidence scores. This bottom-up approach does not rely first on the detection of the person, but rather the different body parts. It has proven to perform well even under self-occlusions situations and to achieve real-time performance.

Objects in the scene are detected by Single Shot MultiBox Detector (SSD) [18] pre-trained on the MSCOCO dataset [16] featuring 80 objects categories. Categories include everyday-life objects *e.g.*: cup, bottle, microwave, TV, desk, smartphone. This is a famous object detector with up to 3400 citations to this day.

As seen in Section 3.1, we have to set values for the action transition matrix and the standard deviations for each Gaussian probability density. In object-object interaction, they depend on distances between pairs of objects, *e.g.* pour milk in a cup. In the affordance model, they may depend on the distance between head and object, *e.g.* drink water, or also depending on velocity ranges. The transition matrix between actions needs also to be tuned and its size is $n \times n$, n being the number of actions. Manual tuning of a great number of parameters is difficult, and due to the dimensionality of the configuration space, grid search methods are not suitable. Instead, we use a hyper-parameter optimization method: SMAC [7], which has up to 900 citations. It has proven already to increase the performances of computer vision algorithms by tuning hyper-parameters [4, 19]. In order to find the set of parameters values, it requires a measure of the algorithm's performance. SMAC optimizes parameters with respect to a cost function. In our case, we want to find the set of parameters leading to the best results in terms of action classification. Therefore, we compute the performance metric F_1-score, f_{a^i} of each action in A, with n being the number of actions to consider. Thus, the cost function C is defined by equation (6):

$$C = n - \sum_{i=1}^{n} f_{a^i}. \tag{6}$$

Upon each iteration, SMAC builds a model of the cost function, which allows it to find the optimal set of parameters with few evaluations. SMAC takes into account previous evaluation outcomes so that it spends less time on the evaluation of irrelevant parameters values, in contrast to random or grid search methods. We compare the inferred action labels to the ground truth to compute F_1-scores and so the



(a) Front view (b) Side view

(c) Front view (d) Side view

Figure 3. Example views of the dataset CAD-120 [13] on 3(a)-3(b) and Watch-n-Patch [28] on 3(c)-3(d) with associated ground-truth objects in red boxes, OpenPose upper-body skeleton and action labels are in blue.

cost function C.

4. Experiments

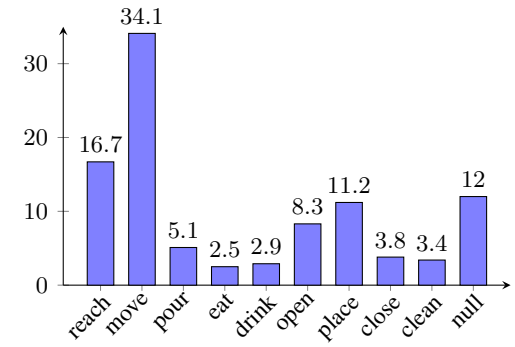
In this section, we present the public datasets and metrics used to evaluate our approach, and then we present and discuss the results in comparison to the literature.

4.1. Datasets and metrics

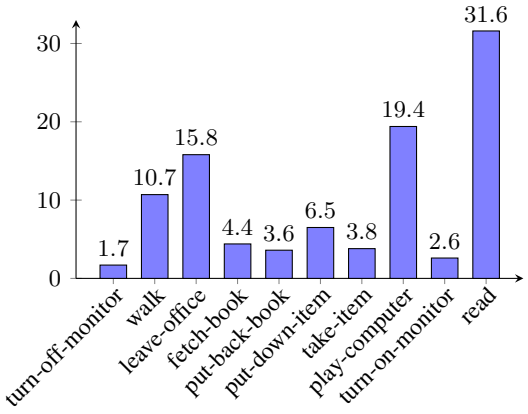
We evaluate our approach on two public datasets in the literature: CAD-120 [13] and Watch-n-Patch [28]. Thus we can make a comparison of our approach with state-of-the-art methods.

CAD-120 dataset [13] consists of 120 RGB-D videos that are performed by 4 subjects. There are 10 activities (*making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, taking food, arranging objects, having meal*). In this dataset, each video sequence represents an activity. As defined in Section 3.1, an activity corresponds to the execution of successive actions. For example, the activity *making cereals* is made of the action classes *reach*, *move*, *pour* and *place*. All the actions of the dataset and their unbalanced distribution, expressed in percentage of corresponding frames, are shown on the bar chart of Fig. 4(a). This dataset offers front and side views, see Figs. 3(a) 3(b).

Watch-n-Patch dataset [28] is split into two environments: *kitchen* and *office*. The environment *office* consists of 196 videos recorded in 8 different offices with 10 different actions (*reading, walking, leave-office, fetch-book, put-back-book, put-down-item, take-item, play-computer, turn-on-monitor, turn-off-monitor*). Distribution in the dataset of the 10 actions is on the bar chart of Fig. 4(b). Each video in the dataset contains 2-7 actions where the actor interacts with different objects in a cluttered background. Also, it offers a variety of camera viewpoints: front and side views are illustrated in Figs. 3(c) 3(d).



(a) CAD-120 [13]



(b) Watch-n-Patch [28]

Figure 4. Action class distribution in datasets CAD-120 [13] and Watch-n-Patch [28]. Corresponding percentage of frames in the dataset to each action label on the y-axis.

In the literature, the approaches are evaluated and compared using two main metrics: F_1-score and accuracy as micro precision and recall. The F_1-score measures accuracy as the harmonic mean of precision and recall. As our approach only outputs one label per frame, in this case the accuracy represents the ratio of correctly labeled frames, also called frame-wise accuracy.

Then, approaches in comparison often provide details about macro precision and macro recall. The difference in the computation between micro or macro depends on the consideration of whether classes in the dataset are balanced or not. Those metrics are computed with or without the temporal action segmentation depending on the nature of the approach.

For a fair evaluation on CAD-120 [13], we use a 4-fold

Table 1. Precision (P), Recall (R) and F_1-score (F1) averages over all the classes on CAD-120 [13] dataset using ground-truth segmentation.

Approach	P	R	F1
GPNN [23]	0.88	0.86	0.87
HELK [6]	0.83	0.82	0.82
S-RNN [9]	N/A	N/A	0.83
QHWZ [22]	0.71	0.68	0.69
Ours	**0.84**	**0.80**	**0.82**

cross-validation with a new human subject in each fold, in a similar way as [6, 13, 22, 23]. We use the SMAC framework to tune the hyperparameters in the training folds, leaving one fold for testing.

4.2. Evaluations and discussion

Hereafter, we compare our action detection proposal against state-of-the-art approaches. Our method is originally developed to deal with online action detection and action transitions. We evaluate ourselves in two different settings: with and without ground-truth action segmentation. First on CAD-120 dataset, then on Watch-n-Patch dataset.

As our approach outputs one prediction per frame and not over a sequence, we slightly adapt it as follows. As time bounds of the different actions in the sequence are known, we use a weighted-vote to find the predicted class over this time segment. Average Precision, Recall and F_1-score are derived from the action class distribution in Fig. 4(a) and confusion matrices found in the original publications when available. Results with ground-truth action segmentation on CAD-120 [13] are summarized in Table 1. Methods [23, 9] are data-driven and rely on neural networks for training and inference. Whereas [6] relies on the computation of maximum likelihood estimation over graphs and [22] employs stochastic grammars. We show similar results to S-RNN approach [9] with a difference of 0.01 in the F_1-score, without the need of a large-scale dataset with annotations for training. In comparison to HELK [6], we obtain similar average results and they show better performances in the detection of action *reach* and *move*. However, their weakest performances in terms of precision concern actions *eat* and *clean* (< 0.3), which are among the least frequent in the dataset as shown in Fig. 4(a). Whereas we achieve to maintain good performances over those two actions, as we can see on the next evaluation configuration.

Now we consider a more challenging situation where the ground-truth action segmentation is not available. In this case we compute the most likely action of each frame and the results are shown in Table 2. Macro **P**recision and **R**ecall are derived from confusion matrices of the original publications when available. Accuracy is computed as micro precision and recall, taking into account the balance of actions in the dataset presented in Fig. 4(a). In the CAD-120 dataset, actions *reach* and *move* represent more than half of the instances. Therefore, we present precision and recall for a more precise analysis of the performances despite class imbalance. We show an overall improvement of at least 4% in accuracy. This improvement is especially significant regarding actions *place* (resp. *clean*) with $+0.16$ (resp. $+0.3$) in the F_1-score. We have enhanced the original object detections given by the dataset through SSD detections (*e.g.* table or desk). As objects are often placed on the table, to know its position helps to design a more accurate action model associated with the action place.

Table 2. Macro Precision (R), macro Recall (R), and accuracy per action on CAD-120 [13] dataset without ground-truth action segmentation.

Approach	KGS [13]		KS [12]		**Ours**	
Action	P	R	P	R	P	R
reach	N/A	N/A	0.63	0.65	0.67	0.81
move	N/A	N/A	0.30	0.86	0.47	0.68
pour	N/A	N/A	0.93	0.59	0.8	0.73
eat&drink	N/A	N/A	0.92	0.52	0.90	0.79
open&close	N/A	N/A	0.84	0.63	0.84	0.75
place	N/A	N/A	0.66	0.61	0.84	0.76
clean	N/A	N/A	0.46	0.58	0.88	0.83
mean	0.71	0.62	0.75	0.63	0.80	0.76
accuracy	0.68		0.70		**0.74**	

In the confusion matrix of Fig. 5, we observe a strong diagonal and note that one of the major source of error is the high false positive rate of the action *move*. The action *move* is the most frequent one as it precedes most of the other actions (*pour*, *eat-drink*, *place*, *clean*). The false positive rate is due to transitions between move and the following action, being difficult to define precisely.

We now compare our results with the literature using the dataset Watch-n-Patch. This shows the great adaptability property of our framework because the two datasets are quite different in terms of points of view and actions involved. In Table 3, we report the ratio of correctly labeled frames (Accuracy) as in [28, 29]. We observe a strong improvement compared to approaches of the literature, *i.e.* + $\sim 16\%$ in the accuracy. Approach [24] in comparison, infer action labels solely based 2D image features and optical flow. We show that the additional information about objects configuration and their nature enhance action models, thus action recognition performances. The approach KMHIS [11] relies on motion pattern recognition in 2D through differential images. The significant improvement is also due to our 3D modeling of the scene being more robust to changes in camera viewpoints present in this dataset.

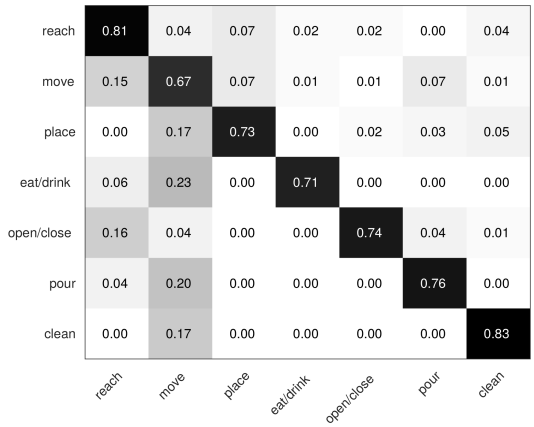


Figure 5. Confusion matrix without ground truth segmentation on CAD-120 [13].

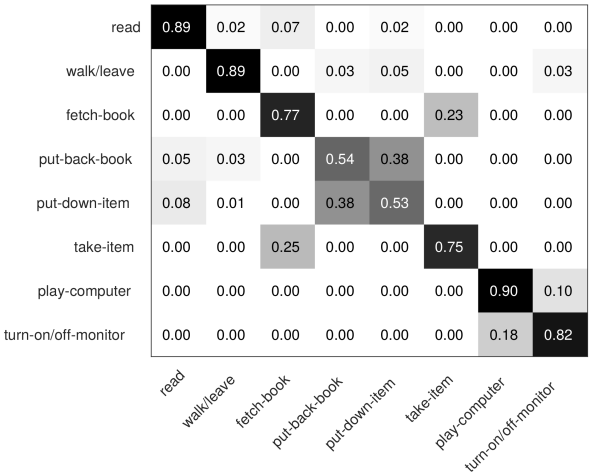


Figure 6. Confusion matrix without ground truth segmentation on Watch-n-Patch [28].

In Fig. 6, we observe good performances over the most frequent actions present in the dataset. The class distribution in the dataset is unbalanced: actions *read* and *play-computer* represent 50% of the labeled frames. While the human pose is similar: the person is sitting in front of a desk with motion in hands, in both cases. They strongly differ in the nature of the object being in interaction: a book or a keyboard. We also achieve a good performance on less frequent actions such as *turn-on-off-monitor*, which represents only 4.27% of the frames. However, in our approach, actions *put-back-book* and *put-down-item* are prone to recognition errors because they involve similar pose and the book or the item is often place on the table. We should modify our models to be more specific towards *put-down-item*.

In Fig. 7 we observe our action detection behavior along the frames of a sequence of CAD-120 [13]. We can also note that actions differ also in execution time length. Some are typically longer such as *move* and *open* in contrast to

Table 3. Accuracy as the percentage of frames correctly labelled of state-of-the-art methods without ground-truth action segmentation on the office environment of Watch-n-Patch [28].

Approach	Accuracy
CaTM [28]	38.5
WBTM [29]	41.2
PoT [24]	49.93
KMHIS [11]	59.75
Ours	**76.1**

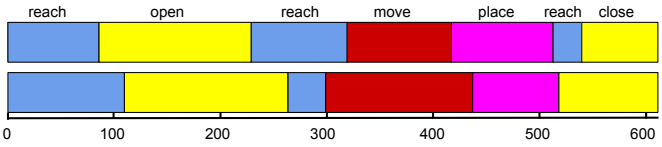


Figure 7. Qualitative result on the sequence *microwaving food* on CAD-120. Frames are on the x-axis. First row is our approach and second row is the labelled ground-truth.

place or *reach*. We have errors on the action transition, on this example we tend to detect the following action too early compared to the ground truth. At the end of action, the probability of the model decreases and the transition matrix favors the probability of the next actions that are likely to occur. This is especially true in CAD-120 where some actions determine the following one *e.g.*: *pour* and *drink* are always followed by *move*.

5. Conclusion

This paper proposes an original framework for online action recognition based on the joint modeling of the human and observed objects in the 3D world. Models are described through Gaussian probability densities which standard deviation parameters are learned using a hyperparameter optimization tool *i.e.* SMAC.

Our joint perception mechanism aims to disambiguate action labeling, *e.g.* the displacement of an object will be found in many different actions class where the nature of the object and its human affordance will strengthen the right action labeling. We don't need a large number of labeled videos to recognize a similar action in another dataset as it might be the case especially for supervised learning techniques, especially CNN ones. Our framework is shown to be more robust to variations in camera viewpoints.

Our evaluations on two challenging public datasets highlight improvements, especially in online action detection. In the future, we will enhance the skeleton observation with the full upper-body, investigate skeleton trajectories, and trajectories predictions based on motion planning. We will also extend the proposal to online activity recognition through plan execution verification.

6. Acknowledgement

This work has been partially supported by Bpifrance within the French Project LinTO and funded by the French government under the Investments for the Future Programme.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*. Springer, 2011.

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[3] C. Chen, R. Jafari, and N. Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *ICIP*, 2015.

[4] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conf. on Artificial Intelligence*, 2015.

[5] J. J. Gibson. The theory of affordances. the ecological approach to visual perception, 1979.

[6] N. Hu, G. Englebienne, Z. Lou, and B. Kröse. Learning latent structure for activity recognition. In *Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[7] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc. of LION-5*, 2011.

[8] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2014.

[9] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[11] H. Kataoka, Y. Miyashita, M. Hayashi, K. Iwata, and Y. Satoh. Recognition of transitional action for short-term action prediction using discriminative temporal cnn feature. In *BMVC*, 2016.

[12] H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *International Conference on Machine Learning*, 2013.

[13] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[14] I. Laptev, B. Caputo, et al. Recognizing human actions: a local svm approach. In *ICPR*, pages 32–36. IEEE, 2004.

[15] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[17] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*. Springer, 2016.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 2016.

[19] C. Maurice, F. Madrigal, and F. Lerasle. Hyper-optimization tools comparison for parameter tuning applications. In *AVSS*, 2017.

[20] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3):90–126, 2006.

[21] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013.

[22] S. Qi, S. Huang, P. Wei, and S.-C. Zhu. Predicting human activities using stochastic grammar. In *Int. Conference on Computer Vision (ICCV), IEEE*, 2017.

[23] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph graphparsing neural networks. In *ECCV*, 2018.

[24] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *CVPR*, 2015.

[25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2013.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2015.

[28] C. Wu, J. Zhang, S. Savarese, and A. Saxena. Watch-n-patch: Unsupervised understanding of actions and relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] C. Wu, J. Zhang, B. Selman, S. Savarese, and A. Saxena. Watch-bot: Unsupervised learning for reminding humans of forgotten actions. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[30] L. Xia, C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.