Early gesture detection in untrimmed streams: A controlled CTC approach for reliable decision-making

William Mocaër, Eric Anquetil, Richard Kulpa

Early Gesture Detection in Untrimmed Streams : A Controlled CTC Approach for Reliable Decision-Making

William Mocaër[a], Eric Anquetil[a], Richard Kulpa[b]

[a] *Univ Rennes, CNRS, INSA Rennes, IRISA - UMR 6164, Rennes, F-35000, France*
[b] *Univ Rennes, Inria, M2S, Rennes, F-35000, France*

Abstract

This paper focuses on the problem of online action detection for interactive systems, with a special emphasis on earliness. Online Action Detection (OAD) refers to the challenging task of recognizing gestures in untrimmed, streaming videos where the actions occur in unpredictable orders and durations. To address these challenges, we present a skeleton-based system for OAD incorporating a decision mechanism to accurately detect ongoing gestures. This allows us to provide instance-level output, achieving a high level of stream understanding. This mechanism relies on a novel Connectionist Temporal Classification (CTC) loss design that restricts the path possibilities according to the action boundaries. We also present a mechanism to tune the trade-off between accuracy and earliness according to the needs of the interactive system using a weighted label prior. This system includes a 3D CNN network, referred to as DOLT-C3D, exploiting the spatial-temporal information provided by the euclidean skeleton representation. We extensively evaluate our approach on eight publicly available datasets, demonstrating its superior performance compared to state-of-the-art methods in terms of both accuracy and earliness. We also successfully applied our approach to early 2D gestures detection. Furthermore, our system shows real-time performance, making it a suitable choice for interactive systems.

Keywords: Online Action Detection, Gesture Recognition, Early Recognition, 3D CNN, CTC
2000 MSC: 68T05, 68T10, 68T30

1. Introduction

Gesture interaction has become an essential component of many human-machine interactive systems. These systems require real-time detection and recognition of user gestures. Gestures follow one another without any prior information on the temporal location of the gesture, which makes recognition more complex in this context. In this study, we tackle the Online Action Detection (OAD) task, which involves analyzing an untrimmed video stream to detect and recognize gestures in real time.

In the context of interactive systems, a high level of understanding and decision-making capability is essential when it comes to detecting human actions.

However, most OAD approaches produce a result at the *frame level*, with no overall view of the action performed. This leads to a low-level understanding of the sequence and offers little guarantee of the consistency of action prediction over time. Moreover, some gestures may have common beginnings, making it impossible to recognize the action correctly on these first frames. Therefore, a system attempting to produce a frame-by-frame response would necessarily make errors in these first frames, as illustrated in figure 1. In the context of human-computer interaction, it is very important to make as few recognition errors as possible so as not to produce undesirable commands in the application, even if this means not giving a decision in the first few frames. Few works address the OAD task as such a decision-making problem.

It is therefore essential to develop OAD systems that produce output at the *instance level*. Instance-level output can be represented using either detection bounds or a decision point. The choice of output representation depends on the specific requirements of the interactive system. These bounds, or decision points, must be consistent with the nature of the gestures, as some can be recognized earlier than others.

To clarify when a system should make a recognition decision, Nowozin et al. [1] have defined the concept of "action point". An action point in an action refers to a specific moment when the presence of the action is unambiguous and can be consistently identified in all instances of the action. An example is given in figure 1, where two actions with a similar start cannot be clearly identified until a certain moment, at which point the gesture becomes clearly recognizable. We can associate the action point with the first frame of the clear zone. This point can be difficult to define, since it involves all the classes in the set considered.

Attempting to predict before the light zone, during the "ambiguous zone", is more risky, but in some cases it is possible to succeed in recognizing the gesture correctly. The level of earliness required in interactive applications varies according to the specific requirements and application context. Referring to the figure 1, attempting to predict gestures in the ambiguous zone may result in recognition systems that makes earlier detections, but probably at the expense of overall accuracy. For example, a command-based application that uses gestures to create, modify and move objects in a graphical space would require a high level of accuracy for smooth interaction. Earliness, on the other hand, is not essential in this application. On the other hand, a sports training application, such as boxing, where a virtual opponent has to react to the user's actions, requires the system to recognize the user's attack very quickly

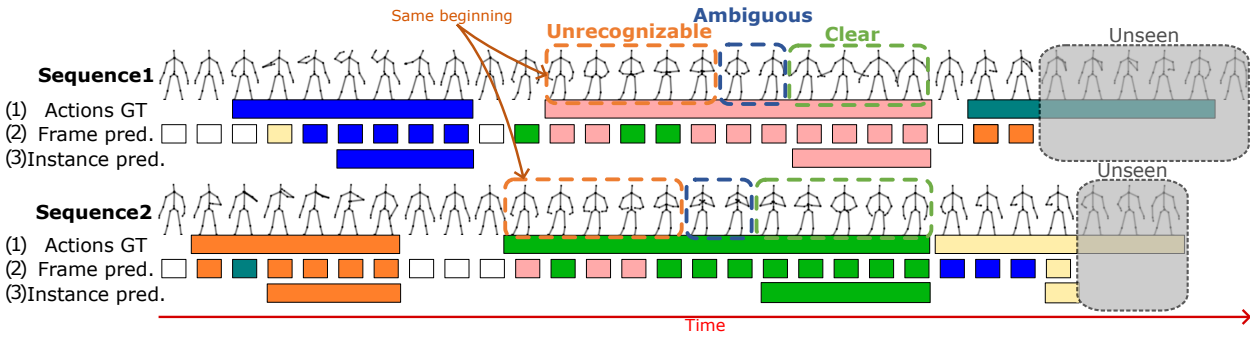


FIGURE 1 – Comparison of the behaviour of a gesture detection system on a data stream at the frame level (2) and at the instance level (3) on different gesture zones. Two sequences of three gestures are represented, the ground truth (GT) bounds of gestures is given (1). The first frames contain the same movements between two actions, so the actions cannot be differentiated. The frame-based system, which aims to recognize each frame, would necessarily make mistakes on these "unrecognizable" frames. An intermediate "ambiguous" zone can be defined at the point where small clues might make it possible to identify the action. The instance-level system would generally make its decision later, during the "clear" zone, but would not make any mistakes.

in order to produce the right defense. The recognition system must therefore be able to achieve different levels of earliness, to make trade-offs between earliness and accuracy according to the specific requirements and constraints of the application.

To summarize, the instance-level OAD task we address in this article has multiple challenges. a) Gestures follow one another in an untrimmed data stream, with no prior information on gesture location. We therefore need to identify the different gesture instances. b) Gestures are not necessarily identifiable from their first frames, so the system needs to be able to avoid making a recognition decision in this area. c) Some applications require this decision to arrive as quickly as possible, while others prioritize detection accuracy to avoid misunderstood commands. A mechanism is therefore needed to adjust the accuracy-earliness ratio. d) The system needs to have a strong recognition capability, yet be efficient enough to run in real time.

We propose an efficient system specifically designed to meet these challenges. To enable effective recognition, we use a new CNN network that features efficient exploitation of the gesture representation. The euclidean representation of the gesture is also designed to be efficient and consistent with the intrinsic design of the CNN. In order to identify each gesture instance, we made a cost function based on the CTC [2], which naturally allows for robust decision making at the instance level. In order to correctly localize the gesture, we guide the CTC by the segmentation of the gesture during learning. Finally, to adjust the balance between precision and earliness, we designed the *weighted label-prior*. This mechanism will allow adjusting the system so that it makes decisions more or less early and precise depending on the chosen intensity.

Our contributions can be summarized as follows :

— We conceived *E-SIM*, a Euclidean representation of the gesture that is independent of execution speed. Constructed from the skeleton, this representation preserves spatial and temporal relationships between joints, making it well-suited for CNNs.

— We built the *Dual-stream Online Long-term Convolutional 3D* (*DOLT-C3D*), an efficient 3D CNN that fully and effectively exploits the E-SIM representation. This network enables the extraction of spatiotemporal features with sufficient context.

— We developed two learning strategies guided by gesture segmentation. Based on CTC, they will enhance the localization of the gesture in the untrimmed stream.

— We introduced the weighted *label prior*, a regularization of the CTC loss that adjusts the precision-earliness ratio to meet various application needs.

2. Related work

2.1. Skeleton-Based Action Recognition

In this work, we present an action detection system based on skeletal information. We rely on the study of Johansson [3], which demonstrated that perceiving and understanding the movement of the human body is possible through the motion of the body's joints, i.e., the skeleton. Additionally, sensors such as the Kinect can effectively provide skeleton data from depth maps [4]. Thus, we think that skeleton information is sufficient for gesture recognition and offers a robust and lightweight approach to online action detection."

Action recognition is a challenging problem in computer vision. Many researchers have addressed this problem, especially on trimmed sequence (one gesture per sequence) and on offline cases (whole clip is available). Many modalities are used in the literature to accomplish action recognition, like RGB videos, depth maps, motion sensors, etc. We focus here on the skeleton modality. The handmade feature approach involves picking important information from the skeleton data, such as the position and angle of joints, and then compute some features using this information to identify the action. Deep learning-based methods are newer and more powerful for recognizing actions. This time, deep networks have to extract by themselves the features from raw data. Nevertheless, several works shown that the way of presenting the raw data to the network, which we call "representation" of skeleton data, can have a decisive impact on performances[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. Three kinds of networks are mainly used in the literature for this task : Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Graph Convolutional Networks (GCN). Recurrent Neural Networks (RNNs) were used a lot for skeleton-based action recognition [15, 16, 17, 18, 19, 20, 21, 22], as they are powerful to recognize temporal patterns. Recently, some transformer-based methods are also emerging for this task [23, 24, 25, 26, 27, 28].

As explain earlier, the way of representing the data is very important, especially for CNN-based methods. Skeleton data representations can be split into two main categories : Joint Coordinate-based and Euclidean Space-based. A summary is available in the table 1. The first category consists of arranging the joints coordinates into a 2D or 3D

matrix. Hou et al. [29] concatenates all coordinates of all joints in one vector for each frame, even if this representation is very compacted and simple, it gets good results when using a 1D CNN. A different arrangement using the same amount of data as Hou et al. has been introduced by Du et al. [30]. It consists of putting the coordinates into a 3D matrix : time $\times$ joints $\times$ joint position (x, y, z values), a 2D CNN can then be applied to this structure. This representation and its variants has been widely used in the literature [5, 31, 32, 33, 9, 8, 34, 22, 35, 12, 13]. A crucial aspect to consider is that the order of joints holds significance in this representation, as the kernel of the first convolutional layer exclusively perceives the neighboring joints in the matrix. Furthermore, there is a limitation in its ability to capture the spatial relationships between joints that are close into the Euclidean space at a time, but not consecutive in the matrix order. In contrast, some works tried to represent the gestures in a 2D or 3D Euclidean space [36, 37, 23, 14]. For example, Duan et al. [14] builds 2D images of the skeleton. Multiple images are generated per frame to localize the placement of each joint and bone. It is worth noting that these images are very sparse but are built as heatmaps. The intensity of the pixels represents the confidence of the presence of the joints or bones in this area. These images over time are used as the direct input for a 3D CNN, with the images of different joints and bones in the channels. The CNN then extracts features using filters sliding across three dimensions : time, X dimension, and Y dimension. This approach improves the previous state-of-the-art results on multiple benchmarks.

TABLE 1 – Summary of the representations used in skeleton-based approaches with CNNs. J : number of joints considered, *time* : number of frames, c : number of output channels, depends on approaches and variants. μ and τ : depends on approaches, τ is always *time*-dependent. X, Y and Z designate here the size of an image. *views* in the number of different viewpoints integrated in the representations.

Type	**Dim.**	**Description**	**Repr. Output Shape**	**Approaches**
Joint Coordinate-based	1D	Time	$time \times (J * 3)$	[29],[38] (3rd stream)
	1D	Time, Component-wise	$3 * (time \times J)$	[11] (1st stream)
	1D	Time, Joint-component-wise	$3 * J * (time \times 1)$	[39]
	1D	Time, Relative distances	$time \times (\frac{(J*J)}{2} + 2 * J * 3)$	[6]
	1D	Joint, Time-wise	$time * (J \times 3)$	[10]
	2D	Time and joints	$time \times J \times 3$	[30, 5, 31, 22, 32, 34, 8, 9, 33, 35, 12, 13]
	2D	Time/joint Grid-arrangement	$(\sqrt{J} * \mu) \times (\sqrt{J} * \tau) \times 3$	[7],[11] (2nd stream)
Euclidean Space-based	2D/3D	Spatial, time cumulated	$views * (X \times Y[\times Z] \times c)$	[37] (2D), [36] (3D)
	3D/4D	Spatio-temporal	$views * (time \times X \times Y[\times Z] \times c)$	[23] (4D, 2nd stream), [14] (3D)

Another way to tackle the spatial correlation problem in the representation of Du et al. is to design a special convolutions mechanism to apply convolutions on the skeleton graph, making use of the semantic structure of the skeleton. The first Spatio-Temporal Graph Convolutional Network (ST-GCN) was introduced by Yan et al.[40] using this idea. GCN-based approaches have brought considerable interest in the field, with numerous publications exploring its use in recent years [41, 42, 43, 44, 45, 46]. Despite their continuous success, GCN-based approaches have been challenged by euclidean space-based methods, including the work of Duan et al. discussed earlier.

2.2. Online Action Detection

Geest et al. [47] defined the Online Action Detection (OAD) task as detecting an action as it occurs, ideally before it completes. Most existing approaches, especially those utilizing RGB modality [47, 48, 49, 50, 51, 52, 53, 54, 55], aim to predict the maximum quantity of frames depicting actions in an online setting. They typically focus on determining the probability of each gesture at each frame and evaluate using frame-level metrics. In contrast, methods employing skeleton modality, such as those in [56, 57, 58, 59, 16, 60], approach OAD by recognizing actions at instance-level. These methods provide one prediction per action instance. In our work, we focus on this instance-level setting, with particular attention to the earliness of detection. In both tasks, a robust temporal modeling is crucial for handling the untrimmed stream of gestures. In earlier approaches, this was primarily addressed by utilizing sliding windows with hand-crafted feature extraction and machine learning techniques [56, 1, 58, 59, 61]. More recently the use of deep learning with RNNs, Temporal Convolutional Network (TCN) and Spatio-Temporal CNNs shows better results.

Fothergill et al. [56] were among the pioneers in addressing this task using sliding windows-based approaches and the skeleton modality. In their work, they extract features from a sliding window and classify using random forests. They also introduced the MSRC-12 dataset for the skeleton modality, with the first protocol for instance-level evaluation. The G3D dataset was introduced in the same year by Bloom et al.[57]. Nowozin et al. [1] defined the concept of "Action Point" mentioned earlier. This important concept can be relevant for evaluating the earliness performance of a system. Knowing the action point frame for each test sample allows to evaluate the accuracy of the prediction by measuring how close it is to this frame. That is the idea of the "Latency-Aware metric" defined in the work of Fothergill et al. However, defining action points can be challenging in the context of human actions, as some actions may be detected early due to previous actions, subtle hints, or external factors. Hence, we introduced an alternative metric named "Bounded Online Detection" (BOD) metric in our previous work [62] on 2D gesture recognition, coupled with the Normalize Time to Detection [63] (NTtoD) to compute the earliness, which can be used for the OAD task.

Boulahia et al. [59, 61] also addressed the modeling of the temporal aspect using sliding windows, but in a different way compared to other methods. Instead of sliding the window over time using a fixed number of frames, they used a fixed amount of cumulated displacement. The features extracted from these windows are independent of the action's execution speed, which is relevant for some actions categories.

RNNs have been widely used for OAD due to their ability to model the temporal dynamics of sequences and provide memory to the system [16, 17, 64, 65, 66, 67, 60, 68, 69]. For example, Li et al.[16] presented a multi-task approach where they jointly learned two tasks : first, a classification task trained with a per-frame loss, and second, a regression task to find the start and end boundaries of each gesture. This allowed the model to have a better generalization capability while considering actions as entities rather than individual frames. In their subsequent work, in addition to using multiple modalities as input, Liu et al. [60] introduced a small TCN module at the start of their network to extract higher-level features such as speed or acceleration, based on the information from local

joint neighbors. The rest of the network is mostly based on LSTM and Fully-Connected layers, processing each frame sequentially and producing the three outputs for each frame : class probabilities, start and end bound probabilities.

Recently, several studies have used CNNs to model the temporal dynamics of gestures for OAD. In particular, Liu et al.[10] and Zhao et al.[52] adopted 1D CNNs (TCNs), instead of RNNs to integrate the temporal information. The frame-level 1D representation vectors of the skeleton were extracted by a CNN-based feature extractor and are used as input of the TCNs. Inspired by the WaveNet network [70], these TCNs employed causal convolutions, which only use information from past frames to make predictions. Dilated convolutions were also used in the work of Liu et al., increasing the network's receptive field to consider more context from past frames.

We have drawn inspiration from different methods discussed in existing studies. In the relevant sections, we will clarify the connections between these approaches and our own methodology.

3. A new Approach for Early Gesture Detection in Untrimmed Stream

3.1. E-SIM : a new Euclidean Speed-Independent Maps Representation for 3D Gestures

First, we will describe our method for representing 3D gestures : E-SIM. This method is based on a representation of the gesture in a Euclidean space, to enable it to be properly exploited by the network.

Inspired by previous works on speed-independent gesture representations [71, 59] and the Euclidean representation proposed by Duan et al. [14], we designed a new representation for skeleton-based systems. Indeed, it has been shown [59] that proposing a speed-independent representation enhances the robustness of a system when speed is not a discriminative factor for recognition. Generally, only the trajectory of a gesture is informative for recognition, and its execution speed is typically not discriminative, although this heavily depends on the nature of the gestures. Regarding the euclidean aspect of the representation, we believe it is particularly relevant when it is being exploited by a CNN, originally designed to process images, which are naturally euclidean. By directly using the 3D positions estimated by devices such as Kinect, we generate 2D heatmaps for each temporal step. The generated heatmaps are projections of the 3D skeleton into two 2D Euclidean space, retaining only the X and Y axes for the first space, and the Y and Z axes for the second space.

Firstly, we need to normalize the skeleton. Given the online context, we cannot normalize using a global bounding box for all images. Instead, we employ the skeleton in each image with a fixed distance over time, such as the arm's length. We normalize the skeleton with a sufficient margin to stretch the arms in all directions. The resulting coordinates fall within the range of 0 to W for the X axis and 0 to H for the Y axis. The skeleton's root is centered for each frame at the position $(W/2, H/2)$.

Secondly, in order to create a speed-independent representation, "chunks" of frames (a chunk being a set of one or more frames) are formed. Within each new chunk, an equivalent amount of displacement θ is achieved. The displacement of all relevant joints $k \in J$ is taken into account to calculate the displacement quantity. We can obtain chunk c from the list of available frames F of size V that have not yet been considered and are chronologically sorted,

as follows :

$$c = \left\{ f_v \in F \middle| \sum_v \sum_{J_k \in J} \|J_{k,f_{v-1}} - J_{k,f_v}\| \leq \theta \right\}, \tag{1}$$

where $\|x\|$ is the Euclidean norm of x, J_{k,f_v} is the vector of 3D coordinates of joint k at frame f_v. An example of a sequence decomposed into chunks is illustrated in figure 2.

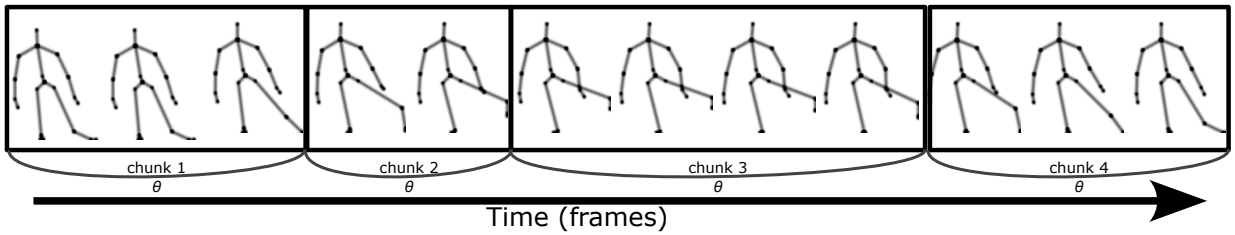


FIGURE 2 – A sequence is decomposed into multiple chunks. Each chunk contains the same amount of displacement θ and may therefore include a different number of frames. A representation is extracted from each chunk.

In addition to being speed-independent, making the input stream more consistent for our network, this segmentation strategy is also a very interesting way to increase the efficiency of our system. During training, much less data is introduced to our model, enabling much faster learning. Furthermore, in the testing context, the system will wait until a sufficient amount of displacement is accumulated before making a prediction, avoiding system saturation.

Thirdly, heatmaps are generated from each chunk. Three types of heatmaps will be drawn for each chunk : joint heatmaps $|J|$, bone heatmaps $|B|$, and a trajectory trace map. We produce one map per joint and per bone to always be able to identify them separately (for example, the 'shoulder' joint map will always be in the same channel of the network).

For the joint heatmaps, we chose to consider only pixels within a distance d from the normalized position of the joint to optimize the process for real-time processing. The pixel intensity depends on the distance to the joints using the same formula as [14] :

$$\boldsymbol{E}^c_{k,i,j} = e^{-\frac{(i-x_k)^2+(j-y_k)^2}{2*\sigma^2}}, \forall k \in \{1, ..., |J|\}, \forall (i,j) \in \mathcal{I}^d_{c,k}, \tag{2}$$

where $\boldsymbol{E}^c$ is the final set of heatmaps for a given chunk, each heatmap is an image of size $W \times H$, x_k and y_k are the coordinates (x, y) of joint k in the last frame of chunk c. $\mathcal{I}^d_{c,k}$ is the set of discrete coordinates (pixels) within a distance d around (x_k, y_k), σ is the variance parameter. Similar to the initial usage, this method of "blurring" the joint positions aims to reduce the impact of pose estimation approximations. The order of joints is not significant (as long as the order remains the same during training and testing). These maps are illustrated at the center of figure 3.

For the bones, using a similar idea, the pixel intensity depends on the distance to the bone segment :

$$\boldsymbol{E}^c_{|J|+b,i,j} = e^{-\frac{\mathcal{D}((i,j),[b_0,b_1])^2}{2*\sigma^2}}, \forall b \in \{1, ..., |B|\}, \forall (i,j) \in \beta^d_{c,k}, \tag{3}$$

where $[b_0, b_1]$ is the segment determined by joints b_0 and b_1, $\mathcal{D}$ is a function calculating the distance between a point and a segment. $\beta^d_{c,k}$ represents the set of discrete coordinates within the bounding box given by the two joints of the

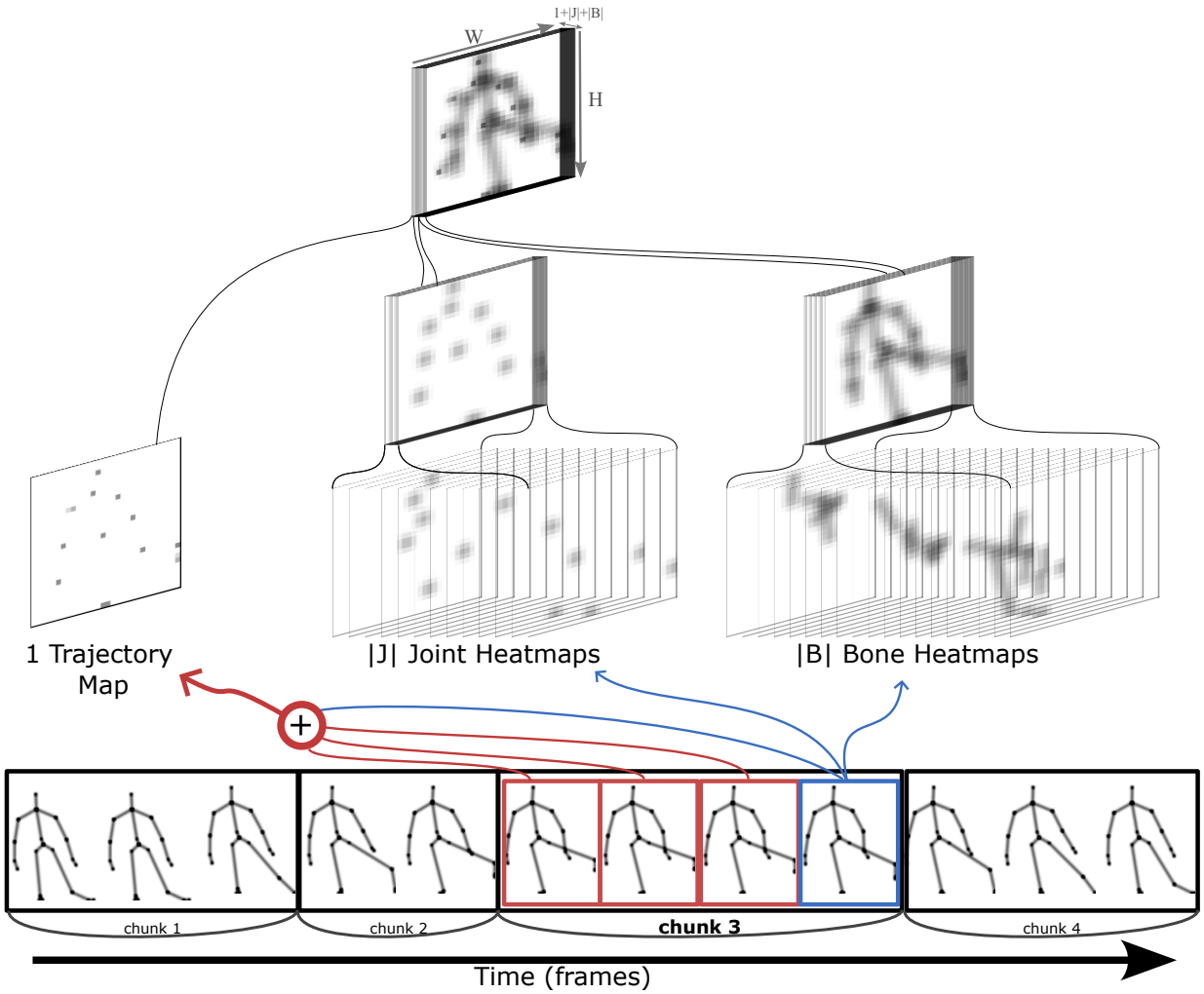


FIGURE 3 – Our generated heatmaps consist of three groups : joint heatmaps and bone heatmaps representing their spatial location, and the trajectory trace map representing the local movement of each joint in the same space. Here, chunk 3 is depicted. Only the last frame of the chunk is used to produce the joint and bone maps, while the entire set of frames in the chunk is used to construct the trajectory map. Only Front-View (FV) is represented here.

bone in the last frame of chunk c, with margins of d units. The maps dedicated to the bones are visible on the right side of figure 3.

An additional map is added to the representation. The aim is to incorporate the temporal information of the trajectory into the representation. Since the joint and bone heatmaps only consider the last frame of the chunk, this additional map bridges the gap between the last frames of chunks by plotting all joint positions from all frames of the chunk onto the same image. The pixel intensity reflects the temporal sequence. Those with a maximum intensity of '1' represent the final position of the skeleton within the chunk. This strategy allows for the reconstruction of trajectories in a single image while preserving information about the temporal order. Since this map will be overlaid with others in the network channels, trajectories will share positions with joint positions. It is unnecessary to differentiate all joints

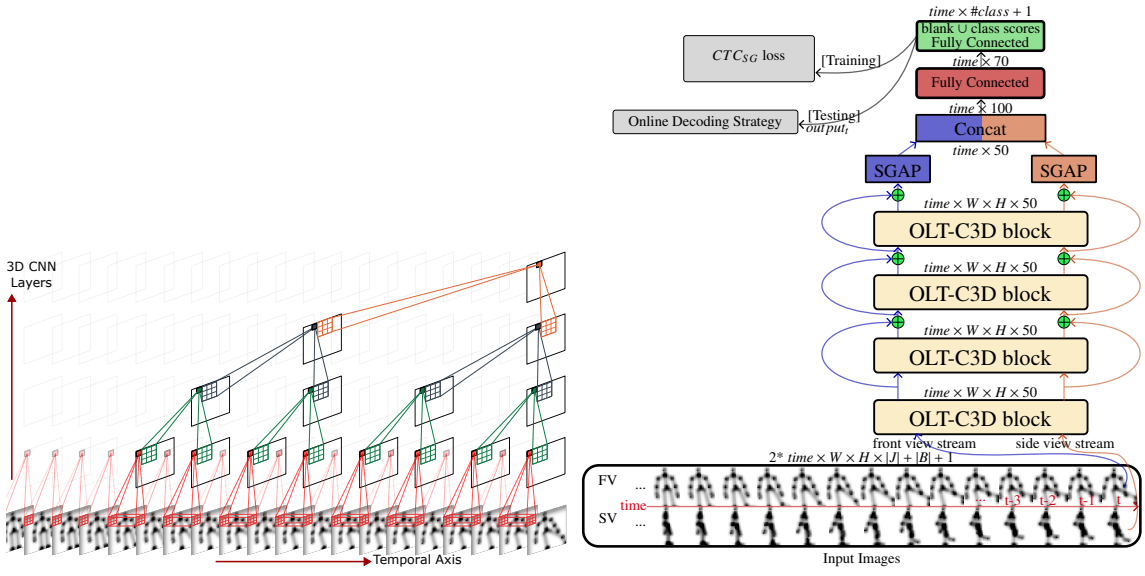
in this map because it is deducible by the network. Moreover, combining all trajectories on a single map allows to learn shared features related to all joints trajectories. Indeed, the same trajectory can be traced by different joints. The distinction of joints is accomplished through the set J. This map is illustrated on the left side of figure 3.

For each chunk, **the front view** (FV, axes X and Y) produces $|J| + |B| + 1$ maps of size $W \times H$, as illustrated in figure 3. By proceeding in the same manner for the projection of **the Side View** (SV, axes Y and Z), we obtain twice this number of maps. The two generated streams, the frontal and side views, will be fed into our DOLT-C3D network.

3.2. An Extended Spatio-temporal Convolutional Neural Network : DOLT-C3D

The Dual-stream Online Long-Term Convolutional 3D network (DOLT-C3D) presented here is primarily inspired by our Online Long-Term Convolutional 3D network (OLT-C3D) described our previous article [62], where it was applied to untrimmed 2D gesture recognition. However, in this section, we extend its application to 3D gesture recognition by introducing the ability to process **two streams simultaneously**, allowing observation from two different perspectives : the Frontal View (FV) and the Side View (SV). Since the gesture actually occurs in 3D, it is essential not to lose the 'Z' dimension. The use of two 2D-projected viewpoints allows us to retain the use of 3D convolutions, losing only a minimal amount of information.

Our network is composed of OLT-C3D blocks. Each OLT-C3D block consists of four 3D convolutional layers with dilation rates of 1, 2, 4, and 8 along the temporal axis. One block is shown in figure 4a. The complete DOLT-C3D network, depicted in figure 4b, comprises four OLT-C3D blocks. For more information on the network, refer to [62].



(a) OLT-C3D Block

(b) Complete DOLT-C3D Network

FIGURE 4 – The presented DOLT-C3D network consists of 4 OLT-C3D blocks and concludes with fully connected layers to produce the number of classes + 1 scores.

As mentioned earlier, our network is dual-stream. It receives inputs from the Front View (FV) and Side View (SV) through two distinct streams, each stream having its set of maps $\boldsymbol{E}$ as defined in the previous section. Both streams share the same OLT-C3D blocks with the same weights to optimize the feature learning. The output features from each stream are then concatenated after a *global average pooling* layer over the two *spatial* dimensions (SGAP), so that the subsequent fully connected (FC) layer can consistently identify the stream origin of the extracted features. The features are then passed through two fully connected layers to produce the output. The network makes a classification prediction for each new chunk. Our architecture can handle sequences of any length during training.

During the testing phase, an online decoding strategy is necessary to achieve instance-level detection based on chunk predictions. This strategy is illustrated in figure 5. At each chunk, we determine the most probable class as the prediction. The detection start boundary is emitted when a class different from the previous one is detected in a chunk. Subsequently, we consider the action to continue until a prediction of a different class is encountered for a chunk. To handle chunks which do not correspond to any gesture or chunks on which the actions are unrecognizable, we use a *blank* class.

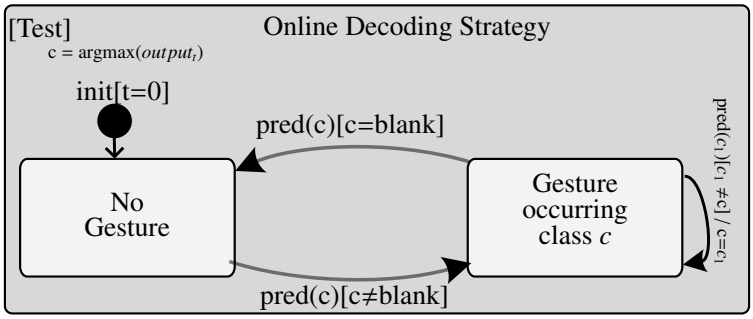


FIGURE 5 – The online decoding strategy is employed. The "blank" is used as a delimiter, and the detected action remains the same instance as long as the prediction remains unchanged. This follows the same strategy as the conventional "greedy" decoding of the CTC.

During training, we need a cost function that is consistent with the instance-level output goal and the online decoding strategy. We choose to use the CTC (Connectionist Temporal Classification) function [2] as it meets this requirement.

3.3. A new Learning Strategy through Segmentation-Guided CTC for Improved Gesture Localization

The CTC (Connectionist Temporal Classification) cost function [2] will serve as the basis for our decision-making mechanism. CTC is generally used to train a model to generate a sequence of labels from a sequential input data, resulting in instance-level output. It has rarely been used in the literature for the OAD task, however we find the work of Molchanov et al. [17]. A study conducted by Zeyer et al. [72] confirms that systems trained with this cost function tend to predict a "peak" in class probability within a limited number of frames (one or two frames), while other frames are assigned the special "blank" label. Moreover, these peaks may not necessarily align with frames corresponding to

the actions. The effect is even more pronounced in online processing, where the CTC tends to predict towards the end of the gesture or even after its completion, as illustrated in the example in figure 6.

The use of CTC ensures the production of instance-level results. However, the imprecise temporal localization behavior mentioned earlier hinders our goal of early detection. In this section, we address the problem of **peak localization** by adding constraints to the paths learned by the CTC function. For this, during training, we use a **Segmentation-Guided CTC** (CTC_{SG}), leveraging the knowledge of temporal segmentation provided by ground truth annotations. The objective is to ensure that the peak occurs between the beginning and the end of the gesture, as illustrated in figure 6.

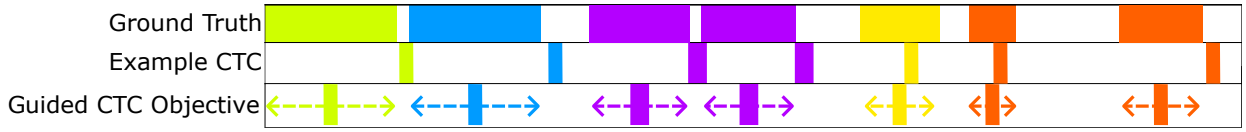
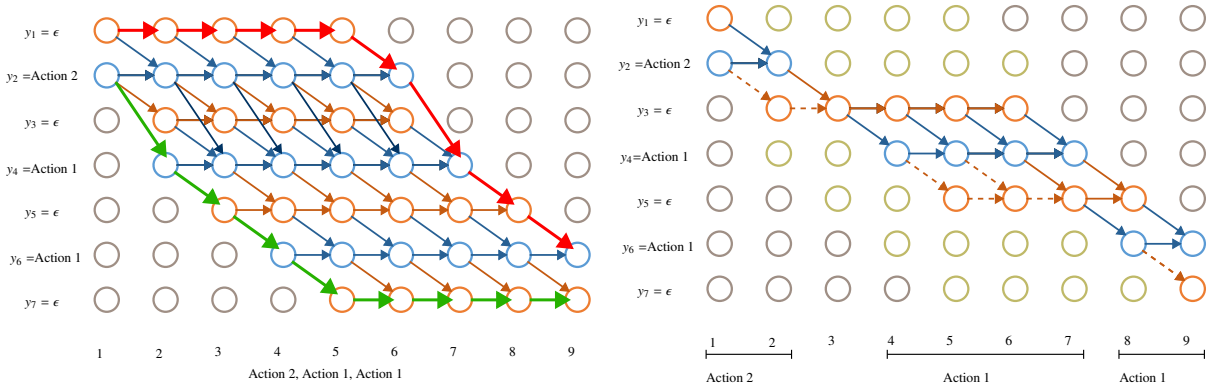


FIGURE 6 – The classical CTC tends to produce predictions with "peaks" that are poorly localized with respect to the gesture. Online, the peak tends to occur lately or even after the gesture completion. The goal of our guided CTC is to teach the system to localize the peak within the bounds of the gesture, between the start and the end.

First, let's define the classical CTC in our context. Let $x = (\boldsymbol{E}^1, \boldsymbol{E}^2, ..., \boldsymbol{E}^C)$ be our input sequence where C is the number of chunks for a given sequence. Let $l = \{1, ..., L\}$ represent the set of possible output labels, and L is the total number of labels. Let $y = (y_1, y_2, ..., y_U)$ be the set of labels where $y_u \in l \cup \{\epsilon\}$, where ϵ represents the empty label. The sequence y consists of the class labels ordered over time, with empty labels inserted before each label and at the end, and U is the number of labels in the sequence (including blanks). The CTC algorithm learns to predict the output sequence $\hat{y}$ based on the input sequence x. For training, it is first necessary to construct a graph representing all possible alignments between x and y. Each node $n_{c,u}$ in the graph corresponds to a possible alignment where the c-th chunk is aligned with the u-th label ($u \in \{1..U\}$). This graph represents the set of possible predictions the system can make on the sequence. For example, to recognize the sequence "action 2, action 1, action 1" within 9 chunks of frames, the model can predict "$21\epsilon1\epsilon\epsilon\epsilon\epsilon$" (green path in figure 7a), "$\epsilon\epsilon\epsilon\epsilon\epsilon21\epsilon1\epsilon$" (red path), or all the intermediate paths represented in figure 7a. The goal of CTC is to allow the system to optimize all paths leading to the correct final sequence (only the order of the labels matters). The transitions in the graph represent valid transitions for these paths. To be valid, a transition can only go from one label to the label that follows it in the order defined by the ground truth or remain on the same label. A transition can also pass through the *blank* (ϵ), and this passage is necessary if two consecutive actions have the same label. We can see the result of the transitions in the figure 7a. More formally, an edge can go from a node $n_{c,u}$ (where c is the index number of the chunk, and u is the index of the action in the sequence) to a node $n_{c+1,u'}$ if the condition C is satisfied :

$$C(n_{c,u}, n_{c+1,u'}) = \begin{cases} y_{u'} \in \{y_u, y_{u+1}\} \\ \textbf{or } (y_{u'} \in \{y_{u+2}\} \textbf{ and } y_{u'} \neq \epsilon \textbf{ and } y_{u'} \neq y_u) \end{cases} . \tag{4}$$



(a) Normal CTC

(b) Our Pruned CTC

FIGURE 7 – Connectionist Temporal Classification (CTC) graphs. (a) the classic CTC graph. The two paths highlighted in red and green designate the two extreme paths which lead to the correct sequence prediction. The green path is the sequence $21\epsilon1\epsilon\epsilon\epsilon\epsilon\epsilon$, while the red path is $\epsilon\epsilon\epsilon\epsilon\epsilon21\epsilon1$. (b) our pruned graph following Soft-SG pruning, and Hard-SG pruning (without dashed transitions). The yellow nodes are the ones which are totally removed from the traditional CTC graph using both pruning strategies.

Until now, all approaches that perform Online Action Detection (OAD) leverage segmentation annotations (start/end) of gestures during training. The classical CTC would only use the order in which actions are performed, regardless of their locations. Thus, we propose two versions of pruning the Segmentation-Guided CTC graph to optimize only relevant paths during training. The goal is to ensure that peaks occur during frames of the actions. We present two versions of pruning in the CTC graph : Soft Segmentation-Guided Pruning (SSG) and Hard Segmentation-Guided Pruning (HSG).

SSG pruning allows the optimization of any path passing through at least one frame of an action **during the action**. This pruning is flexible enough to allow predicting blanks up to the penultimate frame of the action (late prediction) and from the second frame of the action (early detection peak). SSG pruning is similar to the "Local-CTC" used in [73]. We can express the transition condition for SSG pruning as follows :

$$C_{SSG}(n_{c,u}, n_{c+1,u'}) = \begin{cases} C(n_{c,u}, n_{c+1,u'}) \\ \textbf{and } y_{u'} \neq \epsilon \implies s_{u'} \leq c+1 \leq e_{u'} \end{cases}, \tag{5}$$

where $s_{u'}$ and $e_{u'}$ are, respectively, the start and end chunks of action u'. The resulting pruned graph is shown in figure 7b.

The *HSG pruning* removes the paths from the SSG version going from actions to blanks (early detection peak). The objective is to encourage the network to continue predicting this class until the end of the action. This pruning is more relevant with the label prior presented in the next section. The transition condition for HSG pruning is formally

expressed as follows :

$$C_{HSG}(n_{c,u}, n_{c+1,u'}) = \begin{cases} C_{SSG}(n_{c,u}, n_{c+1,u'}) \\ \textbf{and } y_{u'} = \epsilon \implies (s_{u'} \leq c+1 \leq e_{u'} \textbf{ or } (y_{u+2} = y_u \textbf{ and } c+2 = s_{u+2})) \end{cases} . \tag{6}$$

Note that the condition $s_{u'} \leq c+1 \leq e_{u'}$ is always false when there are no blanks between two actions. As two consecutive actions with the same class label need a blank to effectively decode two different actions, the $(y_{u+2} = y_u$ **et** $c+2 = s_{u+2})$ condition ensure to have a path passing by the first frame of the second action since that it allows the last frame of the first action to be a blank. The resulting graph is shown in figure 7b (paths without the dashed transitions).

Our Segmentation-Guided CTC loss $\mathcal{L}_{CTC_{SG}}$ is defined as follows :

$$\mathcal{L}_{CTC_{SG}} = -\log \sum_{\pi \in paths|C_{SG}} \prod_{c:C} p(n_{c,\pi_c}) \quad , \tag{7}$$

where $paths|C_{SG}$ are all the paths leading to the correct final sequence of label, where transitions verify the segmentation guided condition C_{SG} (SSG or HSG). π_c is the label in the path π at the step c. $p(n_{c,\pi_c})$ is the output probability of the model for the label π_c at step c.

Our new guiding strategy enables learning to make decisions during the gesture, unlike the classical CTC. This contribution does not explicitly address the accuracy/earliness balance, which we handle through the weighted label prior, as presented in the next section.

3.4. An Original Weighted Label Prior for Balanced Accuracy and Earliness

In the online context, the earliness of a system refers to its ability to accurately detect and recognize a gesture as soon as possible during its execution. Typically, during the final frames of a gesture, it is easier to classify the action as the system has access to the complete information at that moment. Thus, we expect that the CTC peaks happens on average in the last frames of the gestures. However, our objective is to detect the gestures as early as possible.

One possible approach could involve adding more constraints to the graph. However, since we lack prior knowledge about the temporal localization of the action point of the gestures (i.e., the frame from which the gesture becomes recognizable), adding additional constraints could eliminate relevant paths. Additionally, considering that the bounds annotation in the ground truth may not always be consistent due to variations among annotators, it is necessary to allow for some flexibility in the paths.

Thus, instead of trying to shift the action prediction peak earlier, we present here a way to gradually ”flat” peaks by predicting a larger number of frames in a gesture. Combined with the Segmentation-Guided pruning presented in the previous section, we expect that the first frame classified as the action will arrive earlier. However, as it is risky to make a decision with less information, the system will probably do more false detection from a certain point. There is here a trade-off Accuracy/Earliness that we want to investigate in order to be consistent with the application used. To

address that, we flat the peaks by using a weighted label prior in our pruned versions of the CTC, based on the study of Zeyer et al. [72].

By introducing the label prior term, the contribution of frequently predicted labels, such as the blank label, is effectively down weighted in the loss function. This encourages the model to produce output label distributions that are more uniformly distributed, with less emphasis on frequently predicted labels. Indeed, when examining the construction of the CTC graph (gigure 7a), it can be observed that the blank is overrepresented compared to other classes. It is consistently present between each annotated class. The goal is therefore to **reduce the impact of the** *blank* so that it is less predicted in a sequence. The *softmax label prior* term is defined as follows :

$$P_{prior}(\pi_c) = \frac{1}{C} \sum_{d:C} p(n_{d,\pi_c}) \qquad , \tag{8}$$

which is the average probability of the label π_c on the sequence. The prior can also be estimated on the whole training data [72].

Predicting a blank label on an action frame is not an error, as the action may not clearly be identifiable at that step. The blanks are necessary, especially in the early stages. The traditional CTC has an advantage of producing very stable predictions with fewer errors because the blank label is predicted frequently and is considered safe. By dividing the prediction's score by its prior term (as shown in Equation 9), the labels become perfectly balanced, resulting in actions being as likely as blank. It will also lead to more errors as the blank would be less predicted. To allow tuning this balance between labels, we add a weight Ψ on the label prior. $\Psi = 0$ would lead to the traditional CTC loss, $\Psi = 1$ will totally balance the labels. Adjusting this weight will enable fine-tuning the balance between accuracy and timeliness. This label prior can be applied either to the classic CTC or the guided CTC. In our case, we will couple it with the guided version to constrain the localization of the flattening.

Our final cost function is :

$$\mathcal{L}_{CTC_{SG,\Psi}} = -\log \sum_{\pi \in paths|C_{SG}} \prod_{c:C} \frac{p(n_{c,\pi_c})}{SGrad(P_{prior}(\pi_c)^{\Psi})} \qquad , \tag{9}$$

where SGrad (Stop Gradient) indicates that this part is not optimized by the network.

In summary, our weighting method allows labels to become more or less equiprobable a priori. By making the *blank* less probable, it will be less predicted, and thus the prediction peaks will be flattened. Combined with the localization constraint of the guided CTC, the learning process will encourage predicting "flattened peaks" between the start and end of the gesture. Indirectly, the flatter the peak is, the more it will tend to start early, and thus predict early. This is achieved while maintaining the classical CTC objective of leading to the correct sequence of gestures at the end, ensuring understanding at the **instance level**.

4. Experiments

In the experiments section, we thoroughly evaluate our method. We provide implementation details (section 4.1), describe the eight datasets and metrics used (section 4.2). We also conduct an ablation study to analyze the impact of

different method components (section 4.3). Furthermore, we compare our approach against state-of-the-art methods (section 4.4). This comprehensive analysis offers valuable insights into the effectiveness and robustness of our method across various datasets and evaluation criteria.

4.1. Implementation Details

To reduce noise in the joint positions, a soft online Butterworth filter is applied before computing the representation. To normalize the skeleton, we use the head-root distance as the invariant distance. We consider these joints to be more accurately detected by the Kinect devices than the arm joints. We built our custom CTC implementation based on the CTC implementation of Liu et al. [74]. Additionally, we incorporated a smoothing loss by computing the cross-entropy between the current prediction at time t and the previous prediction at time t-1, multiplied by a weight of 10 to ensure it had a similar magnitude to the CTC loss. Our network hyperparameters are the same for all the datasets : output image dimensions used for E-SIM is $15 \times 15 \times 15$ with distance $d = 2$ (unless explicit mention in ablation study) and θ is set to 3. 13 joints have been selected to be represented in the joints heatmaps, but all the bones are used. Heatmap variance σ is set to 1.3. Regarding the network, we used four OLT-C3D blocks of four 3D CNN layers with 50 filters each, ReLu is employed after the convolutions and the Fully Connected (FC) layer. Dropout is added after each convolution layers (0.2) and after the FC layer (0.3). 70 neurons are used in the FC layer. Spatial maxpooling is used after each convolution layer ($1 \times 3 \times 3$) with padding to keep the same dimensions. The batch size is set to 4 sequences of $\approx$ 200 chunks (40 for G3D as the sequences are much smaller). Mirroring is done as data augmentation. The training is done with Adam optimizer. The network has a total of $\approx$715K trainable parameters. For 2D gesture experiments, we used the representation employed in [62], with $\omega = 2$. The network is used in single-stream mode, with two OLT-C3D blocks of 5 convolution layers. Dropout is applied in all convolutional and dense layers, with a rate of 0.1 for convolutional layers and 0.2 for dense layers. Each convolutional layer learns 30 filters. After the convolutional layers, a dense layer of 100 units is used, with all outputs shared. The network has around 150K parameters. During training, a random rotation is applied to the sequence to augment the data (all images in the sequence undergo the same rotation), following a normal distribution with $\mu = 0$ and $\sigma = 15°$, to improve generalization. Training is performed with a batch of 5 sequences.

Our implementation is available at `https://gitlab.inria.fr/intuidocenlignepublic/OLT-C3D_OAD`. To evaluate our system, we used our evaluation framework available at following address : `https://gitlab.inria.fr/intuidocenlignepublic/evaluation-framework-OAD` for future experiments in the field.

4.2. Datasets and Metrics

To evaluate the performance of our system, we carried out experiments on six databases commonly used in the literature for untrimmed action detection, as well as two 2D gesture databases. A summary of the databases is given in table 2. The data, additional information and splits used are available at `https://www-shadoc.irisa.fr/oad-datasets/`. For the G3D and MAD databases, new test data have been created to avoid gesture order bias.

TABLE 2 – Summary of gesture data sets used in this work. The number of sequences refers to the initial count ; an alternative number is given if different (subset or extended set). Details are given on `https://www-shadoc.irisa.fr/oad-datasets/`.

Nom	**#classes**	**#sequences /used**	**Gestures Nature**	**Device**	**Annotation start/end**	**Annotation Action Point**
MSRC-12 [56]	12	594	Interaction	Kinect V1	✓	✓
MSRC6-Iconic-C4 [56]	6	58				
OAD [16]	10	59	Daily Activities	Kinect V2	✓	✗
G3D [57] (Fighting)	5	30/33	Sport	Kinect V1	✓	✓
MAD [75]	35	40/107	Activities/Sport	Kinect V1	✓	✗
Chalearn [76]	20	680	Interaction	Kinect V1	✓	✗
PKU-MMD [77]	43 (1pers.)	1076/860	Daily Activities	Kinect V2	✓	✗
ILGDB_Untrimmed [62]	21	2831	2D Mono-Stroke	Pen + tablet	✓	✗
MTGSetB_Untrimmed [62]	31	3748	2D Multi-Touch	Touch + tablet	✓	✗

In these experiments we use the metrics Latency-Aware Score [56] and DAP [61] to evaluate online detection quality, as well as NTtoD [63] to evaluate earliness. We also use the BOD [62] metric with parameters $\Delta = 0$ and $canCorrect = False$, enabling us to evaluate our approach with a metric that is more relevant and representative of our context of use.

4.3. Ablation Study

Before making a comparison with state-of-the-art approaches, we first carry out an ablation study to evaluate the performance of the various components of our method. Here, we evaluate our representation strategy presented in section 3.1, then the effectiveness of our CTC-based cost function (presented in section 3.3 and section 3.4) by comparison with variants.

4.3.1. Effectiveness of the E-SIM Representation Strategy

To demonstrate the effectiveness of the E-SIM representation strategy, we conducted experiments on the Chalearn dataset. We first evaluated our representation against that introduced by Duan et al. [14], which transforms each frame into a set of temporal heatmaps. The table 3 shows that our approach processes sequences much faster than the temporal heatmap method. This is because our chunking strategy (the way chunks are constructed) reduces the number of images to be processed, as well as being independent of execution speed. For a dataset with an average of 1370 images per sequence, our chunking strategy produces around 400 images when the displacement quantity θ is set to 3. At runtime, this means that no new images are created or processed until a sufficient amount of displacement has accumulated. This approach avoids overloading the system and makes it well suited to real-world applications. Moreover, our approach defines intensity values locally around key points on the skeleton heatmaps (at a maximum distance d from the key point), instead of iterating over the whole heatmap, which significantly reduces processing time. Inference time is thus reduced from 115 ms per chunk to 25-53 ms, depending on the value chosen for d.

TABLE 3 – Results for different representation variants, using our DOLT-C3D + $CTC_{SG,\Psi=0.1}$. Chalearn dataset, BOD FScore $\Delta = 0, canCorrect = False$. FV is the front view stream, SV is the side view stream. E-TM is the representation without the speed-independent strategy, it is equivalent to the representation used by Duan et al. [14]. "Full map" means that d is not limited. "Processing/seq." is the average processing time (s) for a sequence. "Output size" is the average number of output images per sequence. IT is the inference time (ms) per chunk (for E-SIM) or per frame (for E-TM).

Representation	**FScore ↑**	**NTtoD ↓**	**Processing/seq. (s)**	**Output Size**	**TI (ms)**
E-SIM $d = 2$, FV Only	77.8 ± 0.8	41.3 ± 0.7	**7**	**400**	**17.5**
E-SIM $d = 2$, SV Only	78.7 ± 0.5	41.2 ± 0.9	**7**	**400**	**17.5**
E-SIM $d = 1$	80.7 ± 0.4	40.0 ± 0.8	10	**400**	25.0
E-SIM $d = 2$	81.9 ± 0.4	**39.6** ± 1.2	15	**400**	37.5
E-SIM $d = 3$	**82.4** ± 0.6	**39.5** ± 0.8	21	**400**	52.5
E-SIM Full Map	82.2 ± 0.6	39.7 ± 0.8	46	**400**	115.0
E-TM Full Map [14]	**84.2** ± 0.6	**37.6** ± 0.6	158	1370	115.3

We also experimented with keeping only one projection in our representation, either the X and Y axis (Front View, FV), or the Z and Y axis (Side View, SV). Processing time has been halved but performance in terms of Fscore and earliness is also reduced. Although temporal heatmaps (E-TM, [14] representation) perform best, execution at nearly 9 frames per second with a large number of images to process (3.5 times more than E-SIM), is not suitable for real-time applications. Consequently, the chunk strategy with $d = 2$ or $d = 3$ is a good compromise between performance and speed, and in future experiments we will be using E-SIM with $d = 2$.

4.3.2. Impact of the Guided CTC and the Weighted Label Prior

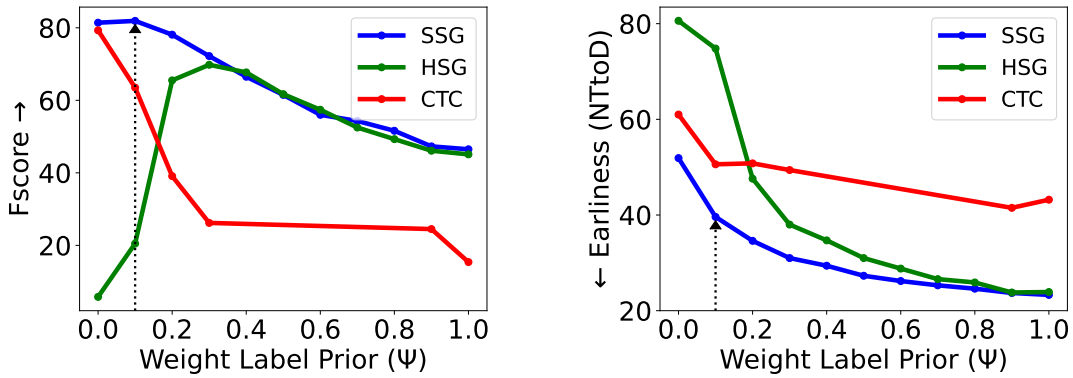
In this study, we analyzed the impact of our pruning strategies and label prior weighting on the performance of our system. To carry out this analysis, we rely mainly on the Chalearn 3D gesture database, but also on six other databases, including the two 2D gesture databases. All the Fscores mentioned in this section correspond to the BOD metric, with $\Delta = 0$ and $canCorrect = False$.

Segmentation-guided CTC compared with Classical CTC. First of all, segmentation-guided pruning gives very interesting results compared with conventional CTC, especially for the SSG version. Table 4 shows that even without any *Label Prior* (when $\Psi = 0$), the system learned with SSG pruning produces much earlier detections (-9%) than the system learned with the classical CTC, in addition to a small gain in Fscore (+2%). These performances confirm that the use of segmentation knowledge during learning is an effective strategy for improving system performance, not only for earliness, but also for Fscore. On the other hand, without any label prior, the HSG version does not deliver satisfactory results (of the order of 15% Fscore), and we have observed that the network has difficulties to converge with this version. Leaving flexibility to the CTC therefore seems preferable in these conditions. We will come back to HSG combined with the label prior later.

TABLE 4 – Comparison of the system trained with different cost functions. The BOD Fscore with $\Delta = 0, canCorrect = False$ is shown with NTtoD (earliness). Chalearn dataset.

Loss	**FScore ↑**	**NTToD ↓**
E-SIM + DOLT-C3D + classical CTC	79.3 ± 2.2	61.0 ± 4.1
E-SIM + DOLT-C3D + $CTC_{SSG,\Psi=0}$	81.4 ± 0.7	51.9 ± 0.7
E-SIM + DOLT-C3D + $CTC_{SSG,\Psi=0.1}$	**81.9 ± 0.4**	**39.6 ± 1.2**

Guided CTC combined with a Low-weighted Label Prior ($\Psi = 0.1$). Then, concerning the guided CTC with the *SSG* pruning version combined with the label prior, we observe in figures 8 (for Chalearn) and 9 (for the other bases) systematically a significant gain in Fscore between $\Psi = 0$ and $\Psi = 0.1$ (+15% for ILGDB, +7% for MTGSetB, +20% for PKU-MMD$_{cv}$, +11% for OAD, +0.5% for Chalearn, +2% for MAD, +9% for G3D). This gain shows that, in general, the weighted label prior ($\Psi = 0.1$) improves the results of guided CTC.



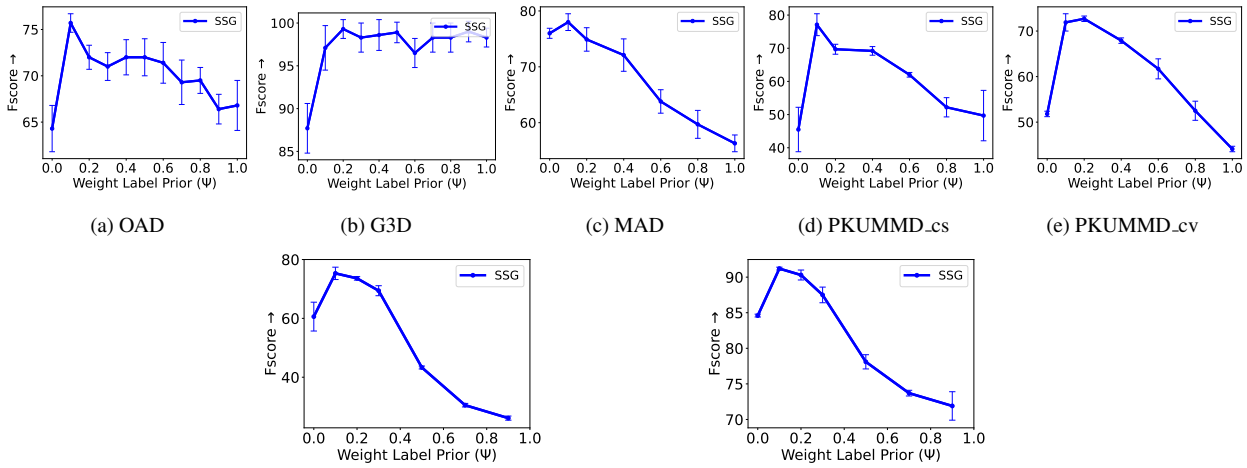
(a) Fscore vs. Ψ

(b) Earliness vs. Ψ

FIGURE 8 – Evolution : a) of the Fscore of BOD ($\Delta = 0$) and b) of earliness related to the weight of the label prior Ψ. Increasing the weight leads to a system that detects gestures earlier. At the same time, Fscore degrades with weight, but some points are more optimal than others. For example, for the SSG version, the best Fscore is obtained when $\Psi = 0.1$ with an interesting earliness performance. Experiments carried out on the Chalearn dataset.

This can be explained by the low recall when $\Psi = 0$. Without the label prior, the impact of the blank in learning is so important that gesture prediction may not be achieved in some cases. For example, for PKU-MMD$_{cv}$, recall rises from 35.5% with $\Psi = 0$ to 58.9% with $\Psi = 0.1$. In addition to the increase in Fscore, there is a significant gain in earliness (from 51.9 % to 39.6 % for Chalearn) for the 3D gesture datasets, as can be seen in figure 10. This result, coupled with the analysis of the qualitative results (figure 11b), allows us to deduce that this is a particular point. Indeed, it shows that detections can be **earlier without reducing recognition quality**. This result is very important, as it means that detections occur closer to the theoretical "action points", these points being the theoretical instant at which gestures are distinguished from each other, and which depends on each gesture class.

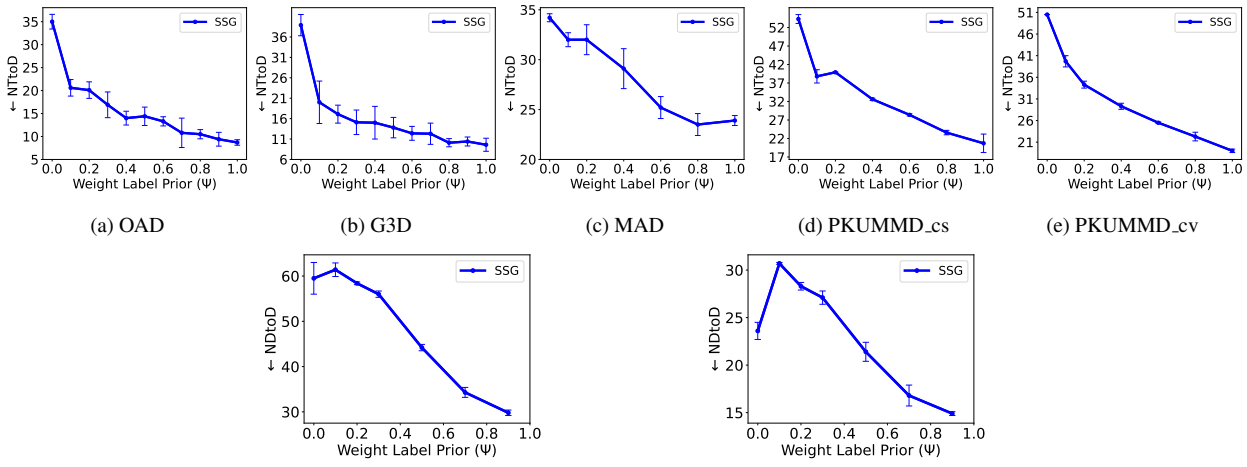
Optimization of the Trade-off between Earliness and Precision ($\Psi \geq 0.1$). We can see in figures 9 and 10 that varying the weight Ψ of the *label prior* allows us to effectively adjust the balance between Fscore and earliness (for SSG with $\Psi \geq 0.1$).



(a) OAD (b) G3D (c) MAD (d) PKUMMD_cs (e) PKUMMD_cv

(f) ILGDB_U. (2D) (g) MTGSetB_U. (2D)

FIGURE 9 – Fscore (BOD $\Delta = 0$) according to label prior weight (Ψ) on 3D gesture databases : a) OAD b) G3D c) MAD d) PKUMMD_cs e) PKUMMD_cv and 2D gesture databases : f) ILGDB_Untrimmed g) MTGSetB_Untrimmed.



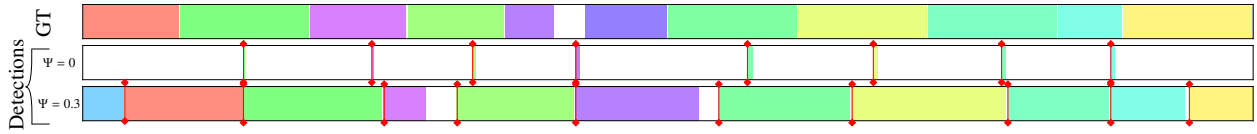
(a) OAD (b) G3D (c) MAD (d) PKUMMD_cs (e) PKUMMD_cv

(f) ILGDB_U. (2D) (g) MTGSetB_U. (2D)

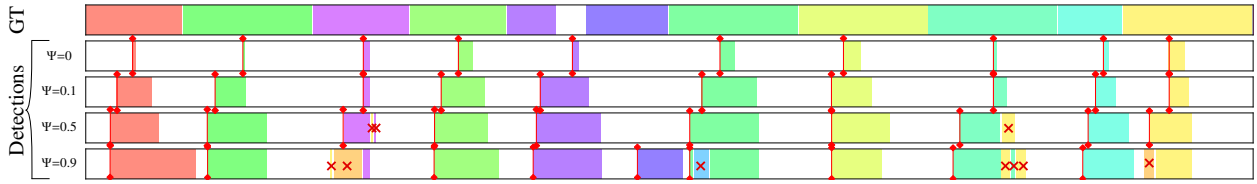
FIGURE 10 – Earliness as a function of label prior weighting (Ψ) on 3D gesture databases : a) OAD b) G3D c) MAD d) PKUMMD_cs e) PKUMMD_cv and 2D gesture databases : f) ILGDB_Untrimmed g) MTGSetB_Untrimmed.

We observed that a higher weight enabled the system to produce earlier detections, but with a generally lower Fscore. This is the same for all 3D and 2D gesture databases. Note that G3D's Fscore does not really decrease, as the score is very close to 100%, with large standard deviations.

The effect of the label prior used to calibrate the earliness/precision ratio is linked the reduced quantity of *blank* predicted, as shown in the qualitative example in figure 11b.



(a) Classical CTC + weighted label prior

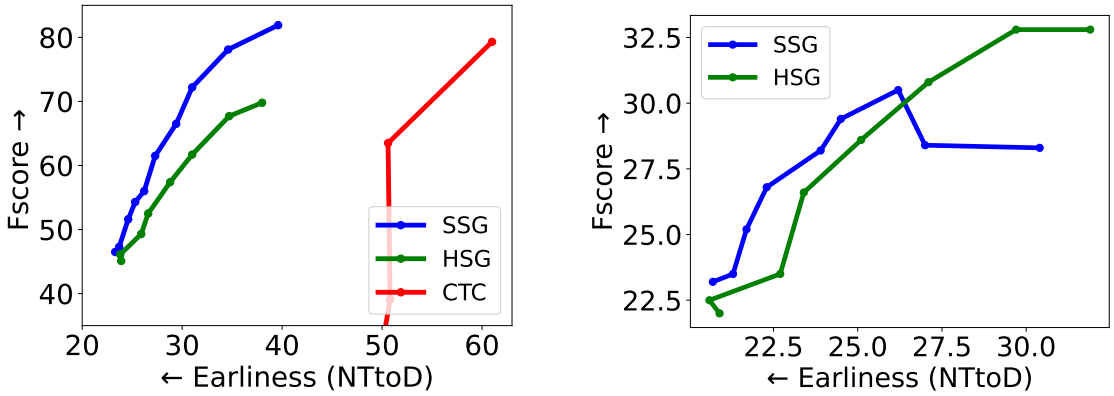


(b) Guided CTC (SSG) + weighted label prior

FIGURE 11 – Example of gesture detection with different label prior weight values (Ψ) for a) classical CTC and b) guided CTC (SSG). The first line (GT) is the annotation (ground truth), with each color designating a class label. Example taken from the Chalearn test database (Sample4). When the weight (Ψ) of the label prior is higher, fewer blanks are predicted, resulting in faster gesture detection for guided CTC (the detection is represented with the red vertical line). However, a higher Ψ value increases the risk of errors (red cross). For the classical CTC, detections do not necessarily come earlier. The result is shown in video at : `https://www.irisa.fr/intuidoc/data/videos/VisualisationSeq00004.mp4`

The presence of the blank allows the system to wait for the gesture to become more clearly recognizable. Predicting fewer blanks in the early stages of a gesture carries a higher risk. When $\Psi = 0.9$, the system makes more errors than when Ψ values are lower. Regarding the classical CTC, when it is combined with the *label prior*, fewer blanks are predicted but detections are shifted in time, which does not lead to earlier detections as shown in figure 11a. This is partly due to the fact that path learning is not constrained by action boundaries in this version. In addition, when actions are temporally close, there is little space left for the detection of the next action. As a result, Fscore performance is poor and convergence difficulties arise. We can see in figure 11a that gestures predicted with $\Psi = 0$ are not really located at the end of annotated gestures (ground truth, GT line) as we would expect with classical CTC. This is because the annotations on this example from the Chalearn database are very "larges", encompassing the gesture with margins, especially at the end of the gesture.

By adjusting the trade-off between earliness and precision using the Ψ weight of the label prior, it is therefore possible to obtain systems with different levels of balance between precision and earliness. In figure 12, each point corresponds to a system state. The performance of these systems can be varied in such a way as to favor precision or earliness, or a compromise between the two. The choice of weight Ψ will help meet the specific requirements of a given application.



(a) BOD Fscore with $\Delta = 0$

(b) BOD Fscore with $\Delta = 0.8$

FIGURE 12 – The use of different weight values for the prior label allows to make systems with varying degrees of precision and earliness. (a) Our SSG pruning methods outperform HSG pruning and the classical CTC for our online detection task. (b) HSG shows interesting results when accurate predictions of end boundaries are required (BOD with $\Delta = 0.8$). Experiments were conducted on the Chalearn dataset with CTC$SSG, \Psi \geq 0.1$ and CTC$HSG, \Psi \geq 0.3$

We also evaluated the performance of HSG pruning, finding that it generally underperformed SSG with Fscore BOD ($\Delta = 0$) for our task, as shown in figure 8. With a low Ψ (≤ 0.2), results are poor compared with the SSG version. With a higher Ψ, both versions perform similarly, but HSG pruning produces generally later detections for a similar Fscore. However, for an application that wants to accurately detect the end of the gesture (BOD score with high Δ), it may be more suitable than SSG pruning in certain configurations, as shown in figure 12b.

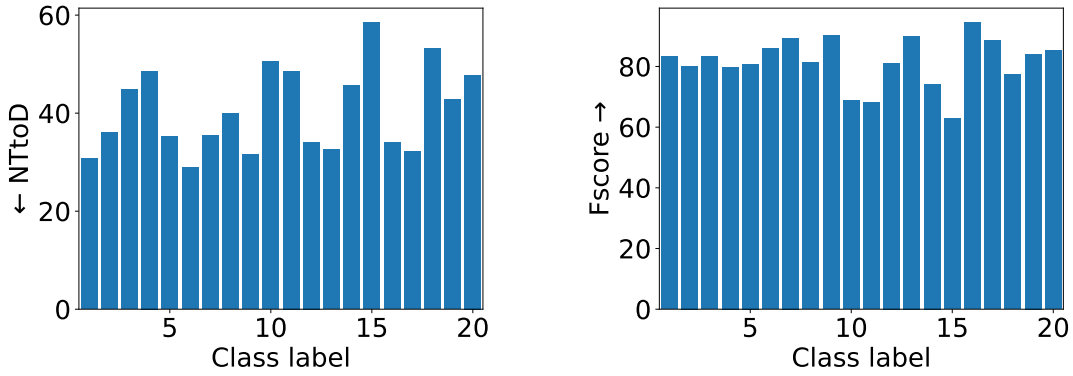
Interestingly, earliness is not the same for all classes, as shown in figure 13. Wide variations can be observed (from 58.5% to 28.9%), showing that the system is able to adapt according to the class of action (some classes are indeed recognizable earlier than others).

A qualitative result at a more granular level than previously illustrated can be seen in figure 14.

Comparison with a Per-frame Cost Function. To demonstrate the effectiveness of our cost function, we compare it to the classical per-frame cost function (cross-entropy) used in most state-of-the-art methods. As shown in Table 5, using the per-frame cost function performs very poorly for the task evaluated with the BOD metric. This cost function takes no account of any form of rejection, since it classifies each frame without any temporal consistency objective at instance level.

TABLE 5 – Comparison of the system trained with a conventional per-frame cost function (cross-entropy) and with our CTC-based cost function. BOD Fscore ($\Delta = 0$) on the Chalearn dataset.

Loss	**FScore** ↑	**NTToD** ↓
E-SIM + DOLT-C3D + Per frame loss	36.7 ± 2.5	**22.7** ± 0.3
E-SIM + DOLT-C3D + $CTC_{SSG,\Psi=0.1}$	**81.9** ± **0.4**	**39.6** ± **1.2**



(a) NTtoD per classe

(b) BOD Fscore ($\Delta = 0$) per class

FIGURE 13 – Results by class on the Chalearn dataset with our $\text{CTC}_{SSG,\Psi=0.1}$. (a) Earliness is highly class-dependent, ranging from 28.9% to 58.5%. (b) Fscores are rather stable around ≈80%.

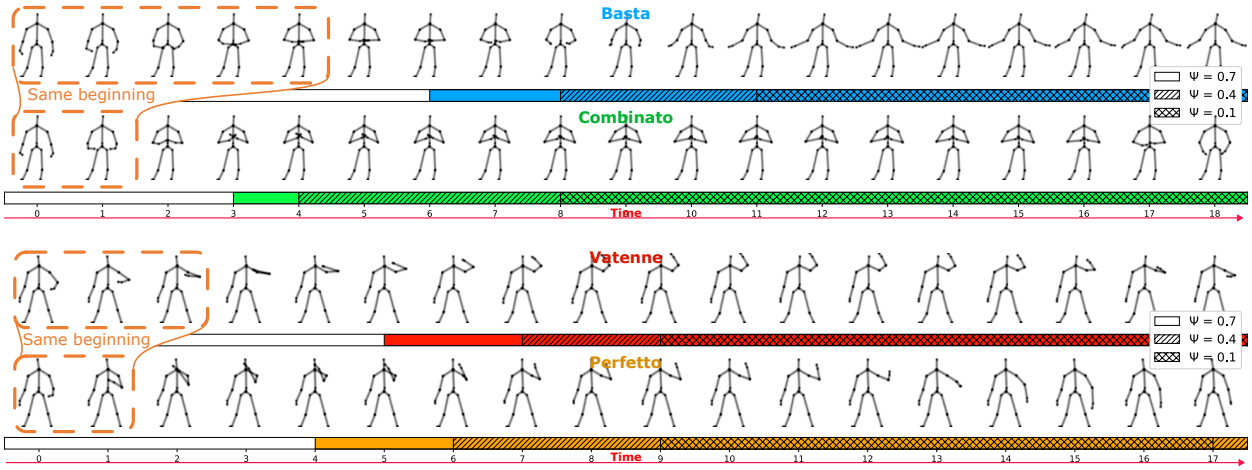


FIGURE 14 – Qualitative prediction results on sequences demonstrating the impact of different Ψ values on prediction timing. The gestures shown from top to bottom are from the Chalearn dataset : *Basta, Combinato, Vatenne* and *Perfetto*. Three different values of Ψ (0.1, 0.4, 0.7) are shown. With $\Psi = 0.1$, we observe that predictions are made when gestures are generally clearly identifiable, implying a later decision. On the other hand, higher Ψ values introduce a higher level of risk on average, with predictions being made at earlier stages of earliness.

4.4. Comparison With State Of The Art

We compared our method with those in the state of the art using the skeleton modality, which focus on instance-level action detection in an online context. For this purpose, consistent with the ablation study, we used : E-SIM with $d = 2$ and CTC_{SSG}, the value of Ψ will be specified for each experiment.

4.4.1. Evaluation of Online Action Detection on MSRC-12 and G3D using Action Point

First, we compare our method with previous approaches using the Latency-Aware Score. This metric evaluates online detection, but without any specific measure of detection earliness. However, as it uses the action point as a

reference for calculating the metric, it makes sense in our context. Two small datasets are evaluated with this metric, G3D (table 6) and MSRC-12 (table 7).

For G3D (table 6), we obtained state-of-the-art results [58, 59], close to 100%. The evaluation focused on the "fighting" categories following a leave-subject-out protocol over 10 folds. Each of the 10 test sets consists of 3 sequences of 5 actions performed by a user.

TABLE 6 – Latency aware Fscore (Δ = 10 frames) based on G3D. Our method achieves results similar to those of the latest approaches.

Method	**FScore**
DFS [78]	91.9
RTMS [79]	92.1
RF [80]	94.8
CAM [58]	97.8
CuDi3D [59]	**98.9**
E-SIM + DOLT-C3D + CTC$_{SG,\Psi=0.2}$	98.3 ± 2.8

For MSRC-12 (table 7), the difference with previous approaches is significant. We obtain superior results in all categories, with 7.4 % of gain in the mean score for all instructional modalities compared to the previous state of the art [81, 82, 59] in this experiment. The protocol used is a cross-subject evaluation of 10 subjects. Each instruction modality is trained and tested separately. A minimum test set is created for each *fold* by taking random subjects from the 30 individuals until all gesture classes are present in the test set.

TABLE 7 – MSRC-12, Latency-Aware Fscore. The instruction modalities are : V-Video, I-Images, T-Text.

Method	RTMS [79]	SSS [83]	ELS [81]	IELS [82]	CuDi3D [59]	E-SIM + DOLT-C3D + CTC$_{SSG,\Psi=0.2}$
V	71.3	55.7	72.6	79.5	84.5 ± 8.0	**90.2** ± 4.9
I	65.6	66.6	67.0	69.2	73.1 ± 12.0	**84.4** ± 5.7
T	52.1	71.3	62.2	63.8	67.3 ± 10.0	**77.2** ± 9.8
V+T	63.5	70.7	79.0	82.3	85.4 ± 7.0	**90.1** ± 3.5
I+T	59.6	73.0	71.1	73.8	75.3 ± 9.0	**80.6** ± 9.9
Average	62.4	67.5	70.4	74.1	77.1	**84.5**

4.4.2. Evaluation on the Online Action Detection Task

Few works have looked at the earliness aspect in an online detection context at instance level. Indeed, the comparison approaches in the previous section were evaluated with metrics that did not reflect the system earliness. Nonetheless, the E-CuDi3D [61] approach evaluated itself with a metric that does evaluate earliness in this context, so we will compare ourselves directly to its score. In order to enable further comparisons with metrics that seem more

relevant to us (BOD and NTtoD), we have implemented the last approach that addresses OAD at the instance level, SM-MT [60].

Regarding the 2D gestures, we compare this approach to our previous approach [62] based on a combination of CTC and SelectiveNet.

Detection to Action Point on MSRC6-Iconic-C4. In this experiment, we compare our method with the evaluation done by Boulahia et al. with the method ECuDi3D [61]. The result is shown in the figure 15. We can see that our method has a much better performance. At the earliest points, ECuDi3D has a little higher fscore, but our system quickly caught up. Our method achieves a final score of 92%, outperforming the ECuDi score of 78% (DAP metric).

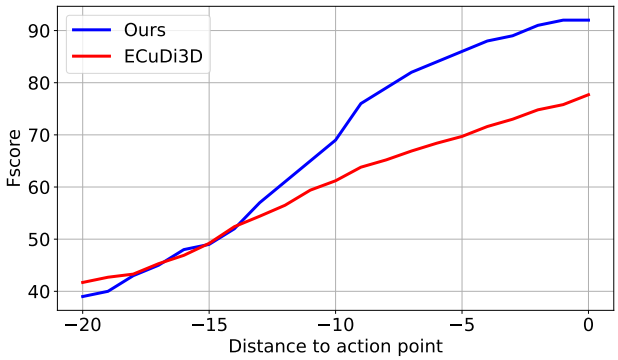


FIGURE 15 – Earliness detection evaluation on MSRC6-Iconic-C4 dataset. The metric used is the Detection to Action Point (DAP) score used in [61]. Our system, E-SIM + DOLT-C3D + CTC$_{SSG,\Psi=0.4}$, shows a better overall performance than the previous work.

Our system shows good ability to classify the gestures at the right moment. In order to have a similar level of risk at the earliest points, we set the parameter $\Psi = 0.4$ in this experiment.

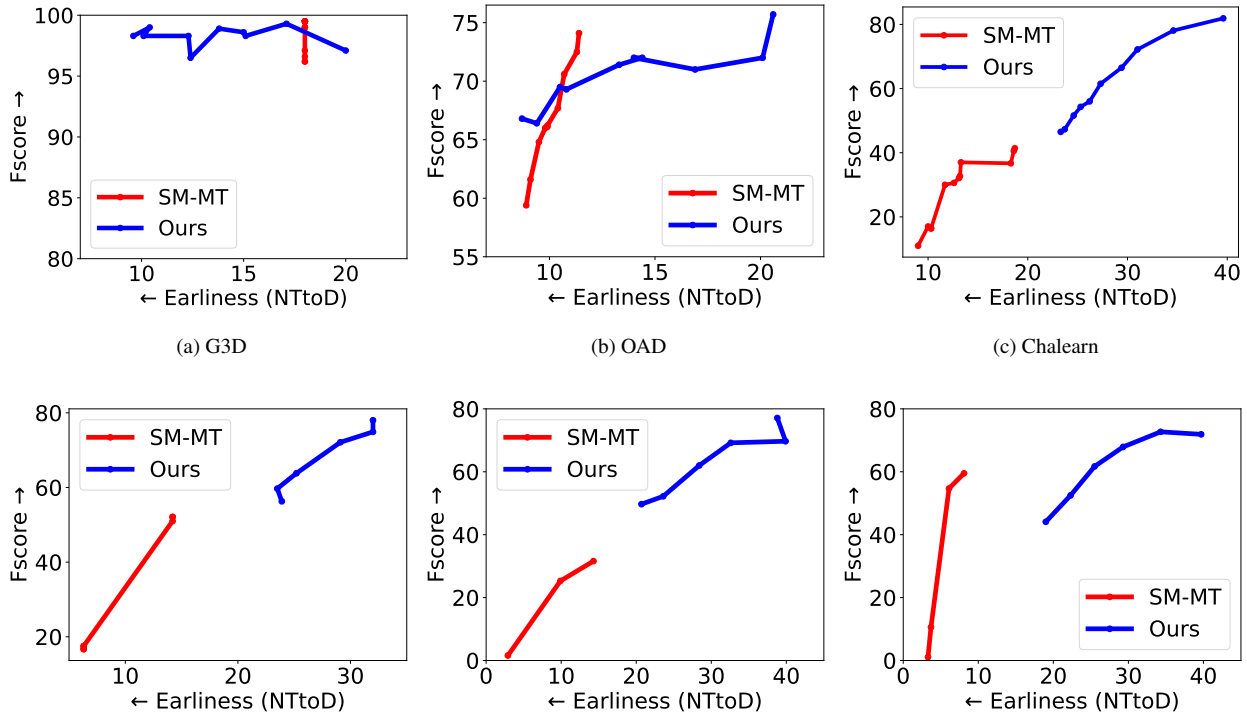
As in [61], experiments were done on the MSRC6-Iconic-C4 subset. The protocol is a 10-fold cross subject, with the same constraint as mentioned earlier.

Detection and Earliness. We also compare our method with a recent state-of-the-art method on the instance-based OAD tasks.

The Skeleton-Modality Multi-Task (SM-MT) approach was introduced by Li et al. [60] in their extension of the JCR-RNN approach [16]. The LSTM-based network jointly learned two tasks : classification and regression. The regression is dedicated to predict start and end bound confidence of the action (supervised with a Gaussian curve centered around the start/end frame) in addition to the class, it allows extracting instance-level gestures. However, trying to predict the start bound with its class is very hard in the online context since multiple actions can have a similar beginning. On the contrary, with our CTC graph design, we allow the system to start predicting the class whenever it deems appropriate. To allow the SM-MT system performs more or less early, we made variations on the start and end confidence thresholds to detect the gestures.

We will evaluate our approach in comparison with SM-MT on 5 databases : G3D, OAD, Chalearn, MAD and PKU-MMD.

First, we evaluated the performance of our approach against the SM-MT approach on **two small datasets, G3D and OAD,** with the metric BOD (*canCorrect*=False, $\Delta = 0$) coupled to NTtoD. The G3D dataset includes 5 classes with low inter-class similarity at the beginning of the gestures (left and right punch, left and right kick, defense). The OAD database contains 10 classes that can also be identified very early on. In this context, our approach demonstrated similar performance to SM-MT, as illustrated in figures 16a and 16b.



(a) G3D (b) OAD (c) Chalearn

(d) MAD (e) PKUMMD$_{cs}$ (f) PKUMMD$_{cv}$

FIGURE 16 – (a) Performance comparison on the G3D dataset : Our approach (E-SIM + DOLT-C3D + CTC$_{SSG,\Psi}$) shows similar overall performance to the SM-MT approach, with the ability to reach earlier points while maintaining close accuracy. (b) Performance comparison on the OAD dataset : SM-MT competes with our approach on earlier points, but we obtain the best maximum Fscore with reasonable earliness. Fscore is Fscore BOD ($\Delta = 0$, *canCorrect* = *False*).

TABLE 8 – Fscore comparison on G3D and OAD datasets. Fscore is the BOD Fscore with $\Delta = 0$. Values for the highest Fscore for each method.

Method	**G3D**	**OAD**
SM-MT	99.5 ± 1.0	74.1 ± 1.9
E-SIM + DOLT-C3D + CTC$_{SSG,\Psi}$	99.3 ± 1.1	**75.7 ± 1.0**

On **G3D**, we can reach earlier points thanks to the weighted *label prior*. However, the fscore remains the same : predicting later is not clearly associated with a better score, which has a large variance (the average variance is around 2.5 %). As gestures can be detected very early due to the nature of the actions, having more or less information to make a decision has no impact on the final score, which is more related to overall recognition ability. Our system reaches areas slightly earlier than SM-MT (≈10% earliness vs. 17% for SM-MT), for a very similar Fscore (around 99%), as shown in table 8. On the **OAD** dataset, the behavior is slightly different. We can see the impact of the weighted *label prior* as the Fscore decreases with earliness. At the point of best Fscore, our approach obtains an Fscore of **75.7%** for a earliness of 20.6%, whereas SM-MT achieves an Fscore of 74.1% for a earliness of **11.4%**. At equal earliness values, two points are interesting : at around 11% of earliness, SM-MT has a 4.6% Fscore advantage (69.5% vs. 74.1%), but at around 9% our approach shows a 7.4% gain (66.8% vs. 59.4%). In short, our approach has similar performance to SM-MT on this dataset.

When the differences between gestures are less obvious on first frames, our method outperforms SM-MT, as we can see from the datasets **Chalearn and MAD** in figures 16c and 16d.

TABLE 9 – Fscore comparison on Chalearn, MAD, and PKU-MMD. The Fscore is the BOD Fscore with $\Delta = 0$. Values for the highest Fscore for each method. Our method (E-SIM + DOLT-C3D + CTC$_{SSG,\Psi}$) obtains the best maximum Fscore.

Method	**Chalearn**	**MAD**	**PKU-MMD (cs)**	**PKU-MMD (cv)**
SM-MT	41.4 ± 2.0	52.1 ± 4.5	31.6 ± 2.3	59.5 ± 2.0
E-SIM + DOLT-C3D + CTC$_{SSG,\Psi}$	**81.9 ± 0.4**	**78.0 ± 1.5**	**77.1 ± 3.3**	**72.7 ± 0.6**

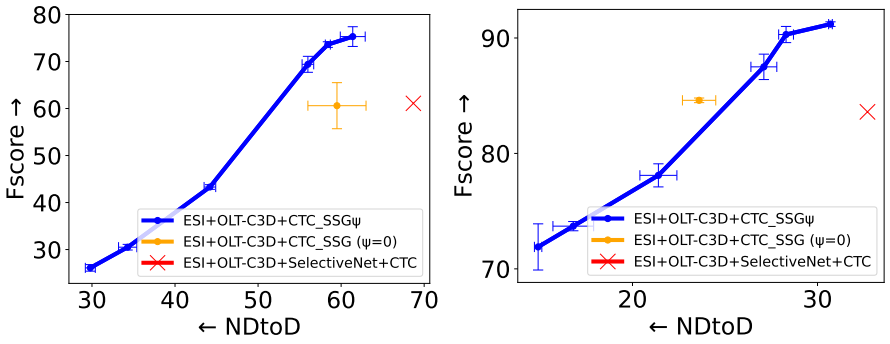
On both datasets, SM-MT reaches areas where the system responds very early, but with a low Fscore (≤ 60%). Our method achieves Fscores (**81.9%** for Chalearn, **78%** for MAD) that SM-MT cannot reach (table 9). The SM-MT detection strategy is designed to give priority to early detection. For example, faced with a similar begin of two gestures, SM-MT classifies them a priori as a given class during the ambiguous zone. For these two datasets, gestures cannot be differentiated from their beginning due to their greater number (20 and 35 classes) and the common beginnings between gestures. This justify the need of a system that is capable of not deciding in certain circumstances. In addition, we need to bear in mind that the NTtoD metric, which expresses earliness, only takes correct classifications into account. The detection score (Fscore of BOD) should be preferred when the difference is significant. Generally speaking, there is little point in having a system that gives answers very early, but which is unreliable.

For the largest dataset, **PKU-MMD** (43 classes considered), our method also outperforms SM-MT (figures 16e and 16f). In terms of Fscore, we obtain respectively **77.1 %** and **72.7 %** for the inter-subject and inter-view protocols at the highest points, whereas SM-MT obtains only 31.6 % and 59.6 %, which are not sufficiently interesting values to allow its use in an interactive system. The good performance of our approach on the inter-view protocol, despite the use of 2D projections of the skeleton in our representation (FV and SV), can be linked to the use of convolution filters whose weights are shared in the two branches linked to the two streams. Filters can therefore be used to extract

common features, probably making them less sensitive to changes in viewpoint. Furthermore, the viewpoint variation in this database is 45° (three distinct views at -45°, 0 and 45°), which does not fundamentally alter the projections.

4.4.3. Evaluation of Online Detection of 2D Untrimmed Gestures

Compared with our previous approach [62], which uses a SelectiveNet + CTC combination (SelectiveNet for addressing earliness, and the classic CTC for decision stability) our new approach (CTC_{SSG} + Ψ) significantly improves performance on both datasets, ILGDB_Untrimmed and MTGSetB_Untrimmed, as shown in figure 17. Here we obtain



(a) ILGDB_Untrimmed (b) MTGSetB_Untrimmed

FIGURE 17 – Comparison on dataset a) ILGDB_Untrimmed and b) MTGSetB_Untrimmed. Our new method with $CTC_{SSG,\Psi}$ is far more interesting than the method combining CTC and SelectiveNet [62], both in terms of early detection and quality. ESI is the representation used in [62]. The Fscore is calculated using the BOD metric $\Delta = 0, canCorrect = False$. Horizontal and vertical bars represent the standard deviation of NDToD and Fscore.

both a much better Fscore (**75.3 %** versus 61.1 % for ILGDB, **91.2 %** versus 83.6 %) but also a better earliness (**61.4 %** versus 68.7 % for ILGDB, **30.7 %** versus 32.7 % for MTGSetB), for Ψ = 0.1 (which gives the best Fscore and the worst earliness).

5. Conclusion

Our work highlights the importance of taking into account the specific requirements of interactive systems when designing online action detection methods. Very few works in the literature have addressed the task of online action detection with a focus on earliness and decision making, which is particularly important in interactive systems.

In this work, we presented a new approach to online action detection in interactive systems. We presented **E-SIM**, a speed-independent representation of gesture in an Euclidean space. We introduced the **Dual-stream Online Long-Term Convolutional 3D (DOLT-C3D)** network, which effectively uses the temporal and spatial information provided by the representation to improve the accuracy of the gesture detection process. Our new cost function, the **Segmentation-Guided CTC**, demonstrated its ability to correctly localize gestures over time, while improving the system's earliness. We also presented an original **Weighted Label Prior** to effectively adjust the trade-off between

accuracy and earliness, making it suitable for a wide range of interactive applications requiring different levels of latency and accuracy. Evaluation on eight publicly available datasets has shown that our approach outperforms state-of-the-art methods in terms of accuracy and earliness. These three components—namely, the E-SIM representation, the DOLT-C3D network, and the guided CTC with or without the weighted label prior—are versatile and independent. They can be integrated into various systems beyond instance-level OAD tasks. Moreover, the enhanced CTC version can be applied to tasks where decision-making capabilities are required. The Segmentation-Guided CTC, especially with the low-weighted prior label ($0.1 \leq \Psi \leq 0.3$), is particularly promising and can be easily used on any neural network-based system that seeks to perform detection (online or offline) in a untrimmed sequence. To conclude, while our method demonstrates promising results in online action detection, the necessity for retraining the system to adjust precision-earliness trade-offs poses a notable limitation. Addressing this challenge, particularly from a user perspective, and exploring more dynamic, adaptive strategies for model tuning will be a key focus for future research.

Acknowledgements

This study is funded by the ANR within the framework of the PIA EUR DIGISPORT project (ANR-18-EURE-0022). Some experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`). This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014187 made by GENCI.

References

[1] S. Nowozin, J. Shotton, Action Points : A Representation for Low-latency Online Human Action Recognition, Technical Report MSR-TR-2012-68, Microsoft Research Cambridge, 2012.

[2] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks, in : Proceedings of the 23rd International Conference on Machine Learning, ICML '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 369–376. doi :10.1145/1143844.1143891.

[3] G. Johansson, Visual perception of biological motion and a model for its analysis, Perception & Psychophysics 14 (1973) 201–211. doi :10.3758/BF03212378.

[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in : CVPR 2011, 2011, pp. 1297–1304. doi :10.1109/CVPR.2011.5995316.

[5] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, A new representation of skeleton sequences for 3d action recognition, in : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[6] F. Yang, Y. Wu, S. Sakti, S. Nakamura, Make skeleton-based action recognition model smaller, faster and better, in : Proceedings of the ACM Multimedia Asia, MMAsia '19, Association for Computing Machinery, New York, NY, USA, 2020. doi :10.1145/3338533.3366569.

[7] J. Liu, N. Akhtar, A. Mian, Skepxels : Spatio-temporal image representation of human skeleton joints for action recognition, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[8] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, W. R. Schwartz, Skelemotion : A new representation of skeleton joint sequences based on motion information for 3d action recognition, in : 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8. doi :10.1109/AVSS.2019.8909840.

[9] C. Caetano, F. Brémond, W. R. Schwartz, Skeleton image representation for 3d action recognition based on tree structure and reference joints, in : 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2019, pp. 16–23. doi :10.1109/SIBGRAPI.2019.00011.

[10] J. Liu, A. Shahroudy, G. Wang, L. Duan, A. C. Kot, Skeleton-based online action prediction using scale selection network, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2020) 1453–1467. doi :10.1109/TPAMI.2019.2898954.

[11] Y. Li, D. Ma, Y. Yu, G. Wei, Y. Zhou, Compact joints encoding for skeleton-based dynamic hand gesture recognition, Computers & Graphics 97 (2021) 191–199. doi :10.1016/j.cag.2021.04.017.

[12] N. Mokhtari., A. Nédélec., P. De Loor., Human activity recognition : A spatio-temporal image encoding of 3d skeleton data for online action detection, in : 17th VISAPP, SciTePress, 2022, pp. 448–455. doi :10.5220/0010835800003124.

[13] H. Qin, J. Cheng, C. Song, F. Hao, Q. Cheng, Structure-preserving view-invariant skeleton representation for action detection, in : 2022 26th International Conference on Pattern Recognition (ICPR), 2022, pp. 3190–3196. doi :10.1109/ICPR56361.2022.9956485.

[14] H. Duan, Y. Zhao, K. Chen, D. Lin, B. Dai, Revisiting skeleton-based action recognition, in : 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 2959–2968. doi :10.1109/CVPR52688.2022.00298.

[15] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks, in : Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, 2016, pp. 3697–3703.

[16] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, J. Liu, Online human action detection using joint classification-regression recurrent neural networks, in : B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 203–220.

[17] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, J. Kautz, Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network, in : The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi :10.1109/CVPR.2016.456.

[18] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, in : Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, 2017, pp. 4263–4270.

[19] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, G. Wang, Skeleton-based action recognition using spatio-temporal lstm network with trust gates, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2017) 3007–3021. doi :10.1109/TPAMI.2017.2771306.

[20] J. Liu, G. Wang, L. Duan, K. Abdiyeva, A. C. Kot, Skeleton-based human action recognition with global context-aware attention lstm networks, IEEE Transactions on Image Processing 27 (2018) 1586–1599. doi :10.1109/TIP.2017.2785279.

[21] H. Wang, L. Wang, Beyond joints : Learning representations from primitive geometries for skeleton-based action recognition and detection, IEEE Transactions on Image Processing 27 (2018) 4382–4394.

[22] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive neural networks for high performance skeleton-based human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (2019) 1963–1978. doi :10.1109/TPAMI.2019.2896631.

[23] L. Shi, Y. Zhang, J. Cheng, H. Lu, What and where : Modeling skeletons from semantic and spatial perspectives for action recognition, 2020. doi :10.48550/ARXIV.2004.03259.

[24] C. Plizzari, M. Cannici, M. Matteucci, Spatial temporal transformer network for skeleton-based action recognition, in : A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, R. Vezzani (Eds.), Pattern Recognition. ICPR International Workshops and Challenges, Springer International Publishing, Cham, 2021, pp. 694–701.

[25] Y. Zhang, B. Wu, W. Li, L. Duan, C. Gan, Stst : Spatial-temporal specialized transformer for skeleton-based action recognition, in : Proceedings of the 29th ACM International Conference on Multimedia, MM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 3229–3237. doi :10.1145/3474085.3475473.

[26] S. Kim, D. Ahn, B. C. Ko, Cross-modal learning with 3d deformable attention for action recognition, 2022. doi :10.48550/ARXIV.2212.05638.

[27] D. Ahn, S. Kim, H. Hong, B. C. Ko, Star-transformer : A spatio-temporal cross attention transformer for human action recognition, in : Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023, pp. 3330–3339.

[28] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, J. Liu, Human action recognition from various data modalities : A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 3200–3225. doi :10.1109/TPAMI.2022.3183112.

[29] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, H. Yang, Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition, in : The European Conference on Computer Vision (ECCV) Workshops, 2018.

[30] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in : 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 579–583.

[31] S. Laraba, M. Brahimi, J. Tilmanne, T. Dutoit, 3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images, Computer Animation and Virtual Worlds 28 (2017) e1782. doi :10.1002/cav.1782, e1782 cav.1782.

[32] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, J. F. Vélez, Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition, Pattern Recognition 76 (2018) 80 – 94. doi :10.1016/j.patcog.2017.10.033.

[33] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu, Y. Zhang, Skeleton-based action recognition with gated convolutional neural networks, IEEE Transactions on Circuits and Systems for Video Technology 29 (2019) 3247–3257. doi :10.1109/TCSVT.2018.2879913.

[34] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, F. Boussaid, Learning latent global network for skeleton-based action prediction, IEEE Transactions on Image Processing 29 (2020) 959–970. doi :10.1109/TIP.2019.2937757.

[35] M. L. Rouali, S. Y. Boulahia, A. Amamra, Simultaneous temporal and spatial deep attention for imaged skeleton-based action recognition, PRIS '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 77–80.

[36] J. Tu, M. Liu, H. Liu, Skeleton-based human action recognition using spatial temporal 3d convolutional neural networks, in : 2018 IEEE International Conference on Multimedia and Expo (ICME), 2018, pp. 1–6.

[37] P. Wang, W. Li, C. Li, Y. Hou, Action recognition based on joint trajectory maps with convolutional neural networks, Knowledge-Based Systems 158 (2018) 43–53. doi :10.1016/j.knosys.2018.05.029.

[38] P.-E. Martin, J. Benois-Pineau, R. Péteri, J. Morlier, Three-stream 3d/1d cnn for fine-grained action classification and segmentation in table tennis, in : Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports, MMSports'21, Association for Computing Machinery, New York, NY, USA, 2021, p. 35–41. doi :10.1145/3475722.3482793.

[39] G. Devineau, F. Moutarde, W. Xi, J. Yang, Deep learning for hand gesture recognition on skeletal data, in : 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 2018, pp. 106–113.

[40] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, Proceedings of the AAAI Conference on Artificial Intelligence 32 (2018). doi :10.1609/aaai.v32i1.12328.

[41] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3590–3598.

[42] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in : 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12018–12027.

[43] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, H. Lu, Skeleton-based action recognition with shift graph convolutional network, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[44] Z. Liu, H. Zhang, Z. Chen, Z. Wang, W. Ouyang, Disentangling and unifying graph convolutions for skeleton-based action recognition, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[45] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, W. Hu, Channel-wise topology refinement graph convolution for skeleton-based action recognition, in : Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 13359–13368.

[46] C. Zhong, L. Hu, Z. Zhang, Y. Ye, S. Xia, Spatio-temporal gating-adjacency gcn for human motion prediction, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 6447–6456.

[47] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, T. Tuytelaars, Online action detection, in : B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 269–284.

[48] H. Eun, J. Moon, J. Park, C. Jung, C. Kim, Temporal filtering networks for online action detection, Pattern Recognition 111 (2021) 107695. doi :10.1016/j.patcog.2020.107695.

[49] Y. H. Kim, S. Nam, S. J. Kim, Temporally smooth online action detection using cycle-consistent future anticipation, Pattern Recognition 116 (2021) 107954. doi :10.1016/j.patcog.2021.107954.

[50] M. Gao, Y. Zhou, R. Xu, R. Socher, C. Xiong, Woad : Weakly supervised online action detection in untrimmed videos, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 1915–1923.

[51] X. Wang, S. Zhang, Z. Qing, Y. Shao, Z. Zuo, C. Gao, N. Sang, Oadtr : Online action detection with transformers, in : Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 7565–7575.

[52] P. Zhao, L. Xie, J. Wang, Y. Zhang, Q. Tian, Progressive privileged knowledge distillation for online action detection, Pattern Recognition 129 (2022) 108741. doi :10.1016/j.patcog.2022.108741.

[53] J. Chen, G. Mittal, Y. Yu, Y. Kong, M. Chen, Gatehub : Gated history unit with background suppression for online action detection, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 19925–19934.

[54] H. Guo, Z. Ren, Y. Wu, G. Hua, Q. Ji, Uncertainty-based spatial-temporal attention for online action detection, in : S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 69–86.

[55] Y. Zhao, P. Krähenbühl, Real-time online video detection with temporal smoothing transformers, in : S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 485–502.

[56] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in : Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, Association for Computing Machinery, 2012, p. 1737–1746.

[57] V. Bloom, D. Makris, V. Argyriou, G3d : A gaming action dataset and real time action recognition evaluation framework, in : 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 7–12.

[58] V. Bloom, D. Makris, V. Argyriou, Clustered spatio-temporal manifolds for online action recognition, in : 2014 22nd International Conference on Pattern Recognition, 2014, pp. 3963–3968. doi :10.1109/ICPR.2014.679.

[59] S. Y. Boulahia, E. Anquetil, F. Multon, R. Kulpa, Cudi3d : Curvilinear displacement based approach for online 3d action detection, Computer Vision and Image Understanding 174 (2018) 57 – 69. doi :10.1016/j.cviu.2018.07.003.

[60] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, W. Zeng, Multi-modality multi-task recurrent neural network for online action detection, IEEE Transactions on Circuits and Systems for Video Technology 29 (2019) 2667–2682. doi :10.1109/TCSVT.2018.2799968.

[61] S. Y. Boulahia, E. Anquetil, F. Multon, R. Kulpa, Détection précoce d'actions squelettiques 3D dans un flot non segmenté à base de modèles curvilignes, in : RFIAP 2018 Reconnaissance des Formes, Image, Apprentissage et Perception, Paris, France, 2018, pp. 1–8.

[62] W. Mocaër, E. Anquetil, R. Kulpa, Early recognition of untrimmed handwritten gestures with spatio-temporal 3d cnn, in : 2022 26th International Conference on Pattern Recognition (ICPR), 2022, pp. 1636–1642. doi :10.1109/ICPR56361.2022.9956529.

[63] M. Hoai, F. De la Torre, Max-margin early event detectors, in : 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2863–2870. doi :10.1109/CVPR.2012.6248012.

[64] Z. Y. Jiyang Gao, R. Nevatia, Red : Reinforced encoder-decoder networks for action anticipation, in : G. B. Tae-Kyun Kim, Stefanos Zafeiriou, K. Mikolajczyk (Eds.), Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2017, pp. 92.1–92.11. doi :10.5244/C.31.92.

[65] R. De Geest, T. Tuytelaars, Modeling temporal structure with lstm for online action detection, in : 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1549–1557. doi :10.1109/WACV.2018.00173.

[66] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, D. J. Crandall, Temporal recurrent networks for online action detection, in : Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[67] M. Gao, M. Xu, L. Davis, R. Socher, C. Xiong, Startnet : Online detection of action start in untrimmed videos, in : 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5541–5550. doi :10.1109/ICCV.2019.00564.

[68] H. Eun, J. Moon, J. Park, C. Jung, C. Kim, Learning to discriminate information for online action detection, in : 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 806–815. doi :10.1109/CVPR42600.2020.00089.

[69] S. Min, J. Moon, Information elevation network for online action detection and anticipation, in : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2022, pp. 2550–2558.

[70] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, K. Kavukcuoglu, Wavenet : A generative model for raw audio, CoRR (2016). arXiv:1609.03499.

[71] Z. Chen, E. Anquetil, C. Viard-Gaudin, H. Mouchère, Early recognition of handwritten gestures based on multi-classifier reject option, in : 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, 2017, pp. 212–217. doi :10.1109/ICDAR.2017.43.

[72] A. Zeyer, R. Schlüter, H. Ney, Why does CTC result in peaky behavior ?, CoRR abs/2105.14849 (2021). arXiv:2105.14849.

[73] T. Zhang, H. Mouchère, C. Viard-Gaudin, A tree-blstm-based recognition system for online handwritten mathematical expressions, Neural Computing and Applications 32 (2020) 4689–4708. doi :10.1007/s00521-018-3817-2.

[74] H. Liu, S. Jin, C. Zhang, Connectionist temporal classification with maximum entropy regularization, in : S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 31, Curran Associates, Inc., 2018.

[75] D. Huang, S. Yao, Y. Wang, F. De La Torre, Sequential max-margin event detectors, in : D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 410–424.

[76] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, H. J. Escalante, Multi-modal Gesture Recognition Challenge 2013 : Dataset and Results, in : Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ICMI '13, ACM, 2013, pp. 445–452.

[77] C. Liu, Y. Hu, Y. Li, S. Song, J. Liu, PKU-MMD : A large scale benchmark for continuous multi-modal human action understanding, CoRR abs/1703.07475 (2017). arXiv:1703.07475.

[78] V. Bloom, V. Argyriou, D. Makris, Dynamic feature selection for online action recognition, in : A. A. Salah, H. Hung, O. Aran, H. Gunes (Eds.), Human Behavior Understanding, Springer International Publishing, Cham, 2013, pp. 64–76.

[79] A. Sharaf, M. Torki, M. E. Hussein, M. El-Saban, Real-time multi-scale action detection from 3d skeleton data, in : 2015 IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 998–1005. doi :10.1109/WACV.2015.138.

[80] S. Baek, K. I. Kim, T. Kim, Real-time online action detection forests using spatio-temporal contexts, in : 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 158–167.

[81] M. Meshry, M. E. Hussein, M. Torki, Linear-time online action detection from 3d skeletal data using bags of gesturelets, in : 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1–9.

[82] S. Wang, Z. Yu, X. Yu, Real-time online action detection and segmentation using improved efficient linear search, International Journal of Computing Science and Mathematics 10 (2019) 129–139. doi :10.1504/IJCSM.2019.098738.

[83] X. Zhao, X. Li, C. Pang, X. Zhu, Q. Z. Sheng, Online human gesture recognition from motion data streams, in : Proceedings of the 21st ACM International Conference on Multimedia, MM '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 23–32. doi :10.1145/2502081.2502103.

William Mocaër received an Engineering degree from INSA Rennes and an MS from the University of Rennes in 2020. He is currently pursuing a Ph.D. degree in the IRISA laboratory. His research interests include handwritten gestures recognition and human action detection.

Eric Anquetil is a full professor at INSA Rennes. He leads research on handwriting, gesture and drawing recognition in the IRISA laboratory. He oversees the "Innovation and Entrepreneurship" mission and the INSA start-up incubator. His expertise lies in AI and man-machine interactivity based on handwritten and gestural commands.

Richard Kulpa is professor at M2S Laboratory in University Rennes 2 and INRIA MimeTIC team. His research is concerned with the use of numerical models of humans in virtual reality and biomechanical analysis to better understand interactions between athletes and their motor performance to propose new sports training tools.