Early Recognition of Untrimmed Handwritten Gestures with Spatio-Temporal 3D CNN

William Mocaër, Eric Anquetil, Richard Kulpa

Early Recognition of Untrimmed Handwritten Gestures with Spatio-Temporal 3D CNN

William Mocaër
Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
william.mocaer@irisa.fr

Eric Anquetil
Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
eric.anquetil@irisa.fr

Richard Kulpa
Univ Rennes, Inria, M2S
F-35000 Rennes, France
richard.kulpa@irisa.fr

***Abstract*—Early recognition of untrimmed handwritten gestures is the task of recognizing as soon as possible gestures drawn in a continuous stream, one after another. This is particularly challenging for multi-touch gestures because it is impossible to know when the gesture has started and finished. For mono-stroke gestures, in an application context where the finger is never removed from the device between gestures, the recognition is even more complex. In this work we present an extension of the Online Long-Term Convolutional 3D (OLT-C3D) network to address the task of early recognition of untrimmed gestures which have been addressed by very few works. To evaluate our approach, we created two synthetic datasets using freely available benchmarks, MTGSetB and ILGDB, simulating the streaming data in two different application scenarios. Furthermore, we propose a new evaluation metric for this specific task. Our approach achieves good performances on the two new datasets and will be a baseline for future works on this challenging task.**

I. Introduction

From the user-interaction point of view, reactive and natural interactions with tactile devices are essential for a successful experience. Gesture interaction allows the user to manipulate naturally the device, but they are often limited to very basic functionalities like zooming, rotating and scrolling. The difficulty of adding new gestures is two folds: recognition accuracy and system reactivity. Increasing the number of gestures increases the probability of having gestures with common beginning. As a consequence, the system cannot predict the gesture from first traces without potentially executing undesirable commands. Waiting until the end of gesture is not an option considering we want a real-time reaction of the device. To apply a zoomming effect when a user make the zooming gesture, we need to detect the gesture from the first instant to produce a direct feedback. To be able to handle such direct manipulations, we need a system capable to recognize very early user gestures, just after that the common part between gestures is passed. Nowadays, it only works for very few well-designed gestures with hand-crafted basic approach that can not be generalized.

Another difficulty in user interaction with gesture is application contexts when gestures are made one after another, then the gestures starting and finishing bounds are not always clear (see figure 1) and can scramble the recognition. This is particularly true with mono-stroke gestures when the finger is never removed from the device between gestures, similarly to a handwritten word but the gestures are completely mixed up spatially. This is also challenging with multi-touch/multi-stroke gestures due to the intra-gesture breaks (when all fingers are removed from the device) between two strokes that cannot be distinguished from an inter-gesture break, so we are not able to tell where the beginning of the gesture is.

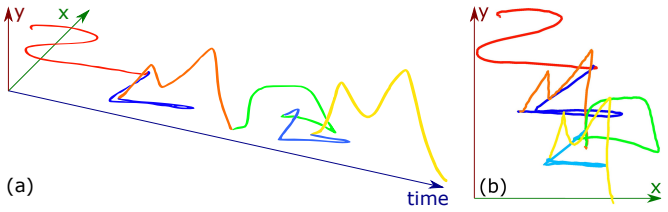


Fig. 1. Example of an untrimmed sequence generated from ILGDB. Each color represents different gestures. In untrimmed context the gestures are chained in the same space. (a): along the three dimensions, x, y and the time. (b): only spatial dimensions.

The task of early recognition of untrimmed handwritten gesture is to recognize the gestures being drawn continuously as soon as possible and before the gesture is fully finished. Few works address this challenging problem, most of them addressed the early recognition of action made by a full human body, either from RGB videos [1] or 3D skeleton videos [2]–[4]. Regarding 2D gesture recognition, some work addressed the early recognition of trimmed gestures [5]–[7], but not in untrimmed context.

To address this problem, we present an approach based on a spatio-temporal 3D CNN called OLT-C3D [7] designed for trimmed gesture recognition. OLT-C3D, coupled with a temporal reject system, is able to postpone a detection to avoid misclassified detection, but it can be unstable because the consistence between frames is not explicitly trained, making it unreliable in untrimmed context. We extend it to address the untrimmed context by regularizing the network with the CTC (Connectionist Temporal Classification) loss [8] which improves the system stability and robustness. To handle this task we defined a new evaluation protocol. We created two artificial datasets to simulate gestures stream, and we propose a new evaluation metric.

The contributions of this paper are summarized as follows:

- We designed a method for the task of early recognition of untrimmed gestures using a spatio-temporal 3D CNN with a temporal reject system. To improve its prediction

stability over time we propose to regularize the network with the CTC loss.
- We propose two speed-independent gesture representation strategies for multi-touch and mono-stroke gestures.
- We built two new challenging datasets of gesture sequences generated to evaluate our system on applicative scenarios and we propose a new metric to specifically evaluate early recognition of untrimmed gestures.

II. Related Works

Some works addressed the task of early recognition of **2D gestures**. Uchida et al. [5] built a system based on multiple frame classifiers, one weak classifier is built per frame. At each current frame, the system combines results from previous and current frame classifiers to determine the current classification. Another classifier combination approach is also used in the work of Chen et al. [6], they designed length-dependent classifiers and a reject system based on the confidence scores and repetition of prediction. Yamagata et al. [9] designed an approach explicitly modeling the trajectory bifurcations between handwritten digits, an LSTM network is used to predict class and the future trajectory. Recently, a new approach based on a 3D Convolutional Neural Network (CNN) called OLT-C3D (for Online Long-Term Convolutional 3D) [7] has been designed to handle long-term visibility without the need of any recurrence layer thanks to temporal dilated convolutions. This approach has a reject system to avoid classification errors in early stages. All these approaches were made for a trimmed context with one gesture at a time, and do not consider the untrimmed context.

Early **3D gesture** recognition from a full 3D human body has more been considered in previous works. Regarding **trimmed** early recognition of 3D gestures, early works used template-based methods to try matching partial gesture sequence [10], [11]. More recently, Wang et al. [12] designed a model trained with teacher-student scheme, two networks are trained to have a close internal representation. The first network (teacher) is able to see the whole sequence while the second network (student) can only see an early part of the sequence. Wang et al. [13] built a network able to predict multiple plausible actions, this is particularly useful in the early stages when the action cannot be clearly identifiable. Also, the model is trained with a weakly supervised strategy by predicting future postures, this helps the network to generalize well. Another way to handle early stages is to define a reject option strategy related to the confidence like some other approaches [6], [7], [14].

Regarding the **untrimmed** case, action prediction of 3D gestures in an untrimmed stream has been addressed by Escalante et al. [15] and Liu et al. [16]. Their networks are able to predict the class based on partial observation, but no strategy to handle early stages are used. The architecture of Liu et al. is inspired by WaveNet [17], using a stack of causal and dilated 1D convolutional layers. SSNet is able to handle a stream in real time, giving a new response to each new frame.

Weber et al. [2] addressed the early recognition problem with a recurrent network. An LSTM network is trained with an additional blank class to represent the inter-gesture frames. Molchanov et al. [1] proposed a convolutional recurrent network (CRNN), the input stream was split into short clips before feature extraction by a CNN. Then, the features are fed into a RNN to extract long-term temporal information. A reject system based on the classifier score is used to handle early stages of gestures. Boulahia et al. [4] used a combination of SVMs trained with a set of hand-crafted features, a complex reject system based on confidence scores has been designed.

From these works, we can notice that the task of early recognition of 2D gesture has not been addressed in the context of untrimmed gestures stream. In the 3D gesture context, most of these methods are trained with a per frame strategy, which can lead to unstable predictions results between consecutive frames, making the method unusable in an application context. Moreover this instability is rarely taken into account in the final evaluation metric since it is often a frame-based metric. In our work, we propose a training strategy which explicitly considers time stability with a CTC (Connectionist Temporal Classification) regularization. Furthermore we present a new metric which hardly penalize unstable predictions over time.

III. Method

In an application context where a gesture is associated with one command, each misclassified detection will lead to an undesirable command execution. To avoid detecting something in the early stages where the gesture is not clearly identifiable we first need an efficient reject mechanism. Secondly, we need to ensure the stability of the predictions to be consistent between consecutive frames.

Our method extends the Online Long-Term Convolutional 3D (OLT-C3D) [7] network to address the early recognition of untrimmed gestures task. First, the online signal of the trace of fingers on the device has to be translated into an image sequence. We designed two new strategies to model the gesture completion in time, inspired by previous ones. The representation is used as the input of the OLT-C3D network trained to handle an untrimmed stream of gestures. A reject option system is used to postpone the detection in early stages. To address the consistency of the detected gesture between frames, an additional output is added to be trained with the CTC loss.

A. Gesture Representation

To choose our gesture representation we need to consider the two different application scenarios which require early recognition in an untrimmed context. In the first scenario, we consider multi-touch gestures made the ones after the others. Between two strokes, all the fingers can be removed from the device, and this is also the case between two gestures. In the second scenario, the finger is never removed from the device, making only mono-stroke gestures, like when we are writing a word with letters.

Like the representation in [7], we choose to represent the gesture being drawn on images to be usable with the OLT-C3D network, at each significant new information we create a new frame representing the gesture at this stage.

We get from the device the online signal, i.e., a list of points, with the timestamp and the positions of the fingers, and we need to convert it into a sequence of images. First, to obtain a speed-independent representation, we resample the gesture using quantity of displacement instead of using the time. Between each new image, the same quantity of displacement (that we called θ) has been drawn on the device, if multiple strokes are done at the same time (multi-touch gesture), then the displacement of all strokes are taken into account to compute the quantity of displacement. We can get the new set of points S from the set of points P not already drawn and ordered by time as following: $S = \left\{ p_t \in P \middle| \sum_{p_{t=1}} \|p_{t-1} - p_t\| < \theta \right\}$.

Note that this resampling strategy is applicable in the online case and if the fingers are not moving, then no processing is required and no new result is given.

Another difficulty is that we cannot guess in advance which size the gesture would be, the user can make the gesture at any scale, but our image has a finite spatial resolution. To address this difficulty, we predefined a scale by advance and if the gesture reaches the border of the image, then we shift the image to the opposite direction to let some space.

To represent the dynamic in the fixed image, we add a second channel on the image to notify the presence of a finger on the device. This second channel is very sparse and contains only ones in positions where the fingers are at, in each image. With this channel, the network can deduce in which direction the stroke is being drawn.

In the previous approach, the gesture was translated into an image sequence where each new image contains the new positions of the fingers with all its previous trajectories, making appear some patterns. In untrimmed context, this is not possible since we don't know when the gesture does start, and we cannot keep all the trajectory of all gestures because the trace will overlap with previous gestures trajectories. Instead, we need a representation strategy compatible with a gesture sequence. This strategy must depend on the context of the scenario described above. We can address differently the multi-touch gestures representation strategy and the mono-stroke gestures one.

1) Multi-Touch Representation Strategy: For multi-touch gestures representation, the trajectory will be completely reset when all fingers have been removed from the device during a very short instant, this can be an inter-stroke moment, or an inter-gesture one. In this way, the trajectory of the gesture is accumulated until strokes done simultaneously are finished. We can add a black image when it happens to notify this event more explicitly to the network. Moreover it allows the network to predict something on this black frame while being sure that the stroke is finished, which is very important for detecting gestures which are subpart of other gestures.

This strategy is not applicable to mono-stroke gestures if the finger is never removed from the device.

2) Mono-stroke representation strategy: For mono-stroke gestures, as there is no identifiable break points, the gesture trajectory is accumulated into an overlapping sliding window ψ. Each image will contain $\psi \times \theta$ quantity of displacement. A big sliding window will lead to noisy images, with pieces of previous gestures, and a too short one will not make appear any pattern on the spatial dimensions. An example of the representation is shown in Figure 2.

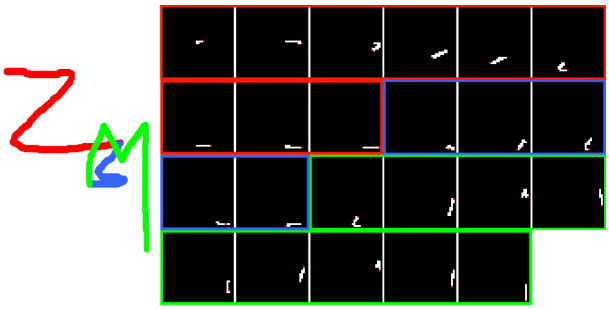


Fig. 2. Example of the representation of a sequence of three gestures. A sliding window ψ of two displacement units is used in this example: each image contains a new piece of information with the previous one.

B. OLT-C3D Architecture with Temporal Reject System and CTC Regularization

The OLT-C3D [7] (Online Long-Term Convolutional 3D) network is composed of a stack of 3D convolution layers. The convolutions are causal: to compute the output of each frame, the future is completely disregarded. This ensures the usability for online applications. The convolutions are temporally dilated in order to increase the receptive field in the time dimension. With two blocks of 5 convolutional layers with increasing dilatation rate, the network can make a prediction while seeing up to 64 previous frames. 64 frames are enough to see at least one full gesture completion. See [7] for more details about the network architecture.

We modify the network to have four outputs: the confidence score (1 output neuron, called g), the classification scores (f), the combination of a blank and the class scores (out_{ctc}) and the auxiliary output (h). These outputs are represented in Figure 3. Three losses are used to train the network.

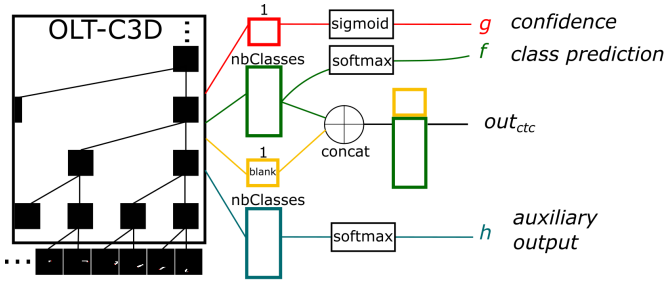


Fig. 3. The network is composed of four outputs. The class prediction output is shared with the CTC output to train a common internal representation.

1) Temporal Reject System: The OLT-C3D network is trained with a per frame loss, originally inspired by the SelectiveNet loss [18], incorporating a confidence output training.

This loss is computed as follows:

$$\mathcal{L}_{(f,g)} = \frac{1}{m}\sum_{i=1}^{m} \ell\left(f\left(x_i\right), y_i\right) g\left(x_i\right) + \lambda\Psi\left(c - \hat{\phi}\left(g\right)\right) \quad (1)$$

where $\Psi(a) = \max(0, a)^2$, λ is a hyperparameter relative to the importance of the coverage constraint (we set it to $\lambda = 32$ like the previous approach). c is the targeted coverage. $\hat{\phi}\left(g\right)$ is the empirical coverage, i.e. the average value of $g(x)$, and ℓ is the cross entropy loss.

During the inference, we will consider that the prediction frame is accepted if g is over a threshold γ, set to 0.5.

The auxiliary output h consists of class predictions like f, but is trained by a traditional per-frame cross-entropy loss $\mathcal{L}_h$.

The temporal reject loss and the auxiliary loss are per-frame losses, and no consideration is given to consistency between consecutive frames. We address this problem with the CTC regularization.

2) Regularization with the Connectionist Temporal Classification (CTC) Loss: The out_{ctc} output of the network is trained with the CTC loss $\mathcal{L}_{ctc}$. Note that the class scores which is part of out_{ctc} is shared with the temporal reject loss.

The CTC loss [8] is used to optimize the alignment between a sequence-level label (i.e., the classes happening during the sequence, without the start/end segmentation bounds) and a per-frame output. The per-frame output is processed in the CTC loss to remove consecutive identical prediction. A blank is also used as a reject to predict none of the available classes.

The use of CTC loss is not necessary for training the network since we have the temporal segmentation of the gestures, we could have trained our network only with the per-frame loss. But CTC brings temporal prediction stability to the network because it requires a good alignment between the sequence-level label and the predicted sequence, and this is very useful for our task. We can notice that the role of the blank in the CTC is relatively close to the one of the confidence score of the temporal reject system. The main advantage of the confidence score compared to the blank is that we can easily tweak it using different parameters (targeted coverage c, λ, confidence threshold γ).

For the final detection, we use the class prediction output f with the confidence score g. The CTC loss is just used to train the internal representation of the network and to smooth the class prediction along the time, we can see it as a regularization strategy.

3) Final optimized loss: The final optimized loss is computed as

$$\mathcal{L} = \alpha\mathcal{L}_{(f,g)} + (1-\alpha)\mathcal{L}_h + \omega\mathcal{L}_{ctc} \quad (2)$$

where we fixed $\alpha = 0.5$ and $\omega = 0.01$ to make the CTC loss magnitude close to the one of the other losses.

IV. Experiments

A. Network details

The images generated by our representation are 40 by 40 pixels, with a quantity displacement θ equals to 4.4 for ILGDB and 1.5 for MTGSetB (once scaled by 0.2 for ILGDB and 0.03 for MTGSetB). The sliding window ψ for the representation of ILGDB is set to 2. Dropout is used in all convolutional and dense layers, with a rate of 0.1 for convolutional and 0.2 for dense layers. Each convolutional layer learns 30 filters. One dense layer of 100 units is used after the convolutional layers, all outputs shared this layer. One additional dense layer, with the same number of units, is used just before the final confidence output layer. During the training, random rotation (following normal distribution with $\mu = 0$ and $\sigma = 15°$) is applied to sequence (the same rotation for all images in a sequence) to improve the generalization. Coverage c of the temporal reject loss is fixed to 0.7 for MTGSetB and 0.3 for ILGDB. The training is done with a batch size of 5 sequences.

B. Synthetic Datasets Generation

To evaluate our approach on the task we generated two datasets from ILGDB [19] and MTGSetB [20]. ILGDB is a mono-stroke gesture dataset containing 21 gesture classes performed by 38 users. These 21 classes are divided into 7 groups of 3 classes, where these 3 classes shared a common begin, making nearly impossible early detection before the bifurcation of these 3 gestures. We generate sequences with between 4 and 8 randomly selected gestures per sequence. The sequence is generated in order that the last point of a gesture is the same point as the first point of the following gesture. Following the original train/test split, 119 sequences are used for training and 210 are used for testing. This dataset is particularly challenging because sequence does not have any breaks so it's very hard to determine the starting and the ending of gestures. Moreover it has few training examples. To tackle the few amounts of data, we generated an augmented dataset using size scaling (5 different scales) and using the same gesture into multiple sequences (each gesture put into 5 sequences). At the end, each training gesture is used 25 times in the sequences, leading to 2621 training sequences. An example of a generated sequence is given in figure 1.

MTGSetB is a multi-touch gesture dataset containing 31 different classes made by 33 users. We built sequences of 4-8 random gestures in order to being unable to differentiate an inter-stroke blank and an inter-gesture blank, each gesture is re-centered according to the previous gesture into the sequence, making impossible a spatial segmentation between gestures. According to the original user-separated train/test split, it leads to 607 gestures sequences for training, and 672 for testing. We also generated an augmented version of the dataset with size-scaling (3 different scales) and using each gesture into 2 sequences. This led to 3076 training sequences.

These two datasets are freely available[1].

[1]Datasets available at:
https://www-intuidoc.irisa.fr/en/mtgsetb-and-ilgdb-untrimmed/

C. Bounded Online Detection (BOD) Metric

We propose a new metric we called *"Bounded Online Detection (BOD) Metric"*, inspired by computer vision detection metrics. The main idea of this metric is to allow only one detection per ground truth bound, conditioning the detection to a certain amount of overlap between the ground truth bound and the detection bound. Any noisy detection will be considered as a false positive. The algorithm allowing to compute the metric is the Algorithm 1.

Algorithm 1 The algorithm to compute the proposed metric.

Inputs: Predictions bounds, Labels bounds ; *Parameters* : canCorrect, Δ.
Sort Predictions and Labels by starting bound.
for all pred **in** Predictions **do**
  $GT^* \leftarrow \underset{GT \in Labels}{\text{argmax}}\ IoU_{st}(pred, GT)$
  if flag(GT^*) = 0 **and** class(GT^*) = class($pred$)
  and $IoU_{st}(pred, GT^*) > \Delta$ **then**
    Add a True Positive ; flag(GT^*) $\leftarrow$ 1
    earliness $\leftarrow \frac{start(pred) - start(GT^*)+1}{end(GT^*) - start(GT^*)+1}$
  else
    Add a False Positive
    if not canCorrect **then** flag(GT^*) $\leftarrow$ 1 **end if**
  end if
end for
Precision $\leftarrow \frac{TP}{TP+FP}$; Recall $\leftarrow \frac{TP}{length(Labels)}$
NDToD $\leftarrow$ average(earliness)

Note that to compute this metric we need the bounds of the predictions. If we have a per frame classification output, we will need a strategy to transform it into a bounded output. This strategy should be compatible with the online context i.e., we should not use future predictions to estimate a starting and ending bound. In our case, the first accepted frame is considered as the start of the gesture, and the next rejection or different class prediction is the end of this gesture.

We designed the IoU_{st} which is a variant of the *Intersection Over Union* measure for online context. Because we don't want to penalize late prediction on this criterion, we compute the overlap from the prediction start bound: $IoU_{st} = \frac{Intersection}{Union_{st}}$ where the $Union_{st}$ is computed from the starting bound of the prediction. A high IoU_{st} characterizes detection which well matches the ground truth bound from the prediction starting bound.

The metric has two parameters: "*canCorrect*" and Δ. *canCorrect* is a boolean, if it is true, it allows the model to correct itself if it has made a detection error on a given gesture. Δ is a value between 0 and 1, and correspond to the minimum IoU_{st} value allowed to consider the gesture as a true positive (TP). For application context which does not require to keep the prediction during the gesture (just a peak), Δ can be set to 0. A value of 1 would mean that the prediction end bound should match exactly the ground truth end bound to be a TP.

Detection is considered as a TP if the ground truth with the maximum IoU_{st} has not been already correctly detected (or falsely detected if $canCorrect = False$), if it is the correct class prediction, and if the IoU_{st} is strictly over Δ. Otherwise it is considered as a *false positive* (*FP*). An example of the application of the metric is shown in figure 4.

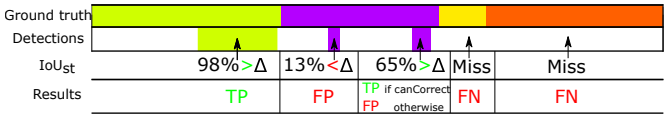


Fig. 4. Example of an application of the Bounded Online Detection (BOD) Metric with Δ=50 %.

Once *Precision* and *Recall* are computed, we can compute the final global micro-averaged *FMeasure* using the traditional computation: $FMeasure = \frac{2*Recall*Precision}{Precision+Recall}$. We also compute the Normalized Distance To Detection (NDToD) used in previous works [6], [7], which measure the earliness of the detection. The NDToD metric is counted only for TP detections.

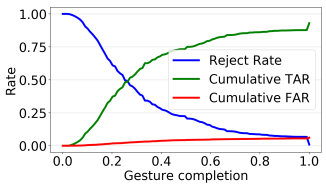
D. Results

1) Trimmed Early Recognition Results: To compare our approach with previous works, we evaluated our approach in a trimmed context. In this context, only the first detection is used because an application would know that only one gesture is being drawn. The comparison with the MTGSetB dataset is shown in table I. Detections happen a little later than [7] but it is well balanced with a significant improvement of the TAR (True Acceptance Rate) and the FAR (False Acceptance Rate) values. Also, almost no gesture is rejected (RR: Reject Rate). This shows particularly the good impact of the CTC regularization, even in trimmed context.

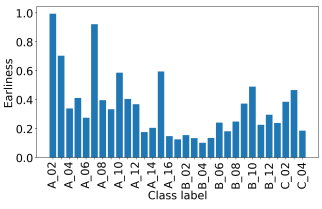
TABLE I
COMPARISON WITH PREVIOUS APPROACHES ON MTGSETB FOR A CLOSE EARLINESS VALUE (NDTOD), TRIMMED EVALUATION.

Method	TAR	FAR	RR	NDtoD
Chen et al. [6]	81.89 %	14.56 %	3.54 %	37.04 %
OLT-C3D [7]	89.25 %	7.24 %	3.51 %	**30.77** %
This work	**93.5** %	**6.44** %	**0.1** %	33.1 %

The figure 5a shows the behaviour of the system according to the normalized gesture completion. 50 % of the gestures were detected before 27 % of their completion. Among them, 95.4 % were correctly classified. As shown in figure 5b, the earliness can vary drastically between gesture categories, from near 100 % for A_02 to less than 15 % for B_04. This depends on the length of the common begins between gestures. Regarding accuracy without earliness, taking the prediction at



(a) Behaviour regarding the TAR, FAR and RR according to the gesture completion ratio.



(b) Earliness per class.

Fig. 5. Performance of the system on MTGSetB (trimmed evaluation).

the last frame for each gesture, we obtain a score of 97.33 % versus 94.45 % for [7].

2) Untrimmed Early Recognition Results: We evaluated our approach in the untrimmed context on the two datasets previously described with the BOD Metric. The results are shown in table II for MTGSetB and ILGDB. The method has been evaluated with different metric parameters combination. For MTGSetB we obtained a FMeasure from 83.6 % to 76.1 % depending on the Δ value, without the possibility to correct the error, and from 88.2 % to 86.8 % if the model can correct itself. These results show that the network is able to predict well until the end of the gesture for this dataset. For ILGDB, the results vary much more depending on Δ, this means that the network is not able to identify correctly when the gesture finishes, this is consistent with the difficulty of this dataset because the gestures are totally chained, and the transitions between gestures are not easily identifiable.

TABLE II
UNTRIMMED EVALUATION ON MTGSETB AND ILGDB (UNTRIMMED VERSIONS) WITH DIFFERENT PARAMETERS OF THE BOD METRIC.

		MTGSetB		**ILGDB**	
canCorrect	Δ	FMeasure	NDToD	FMeasure	NDToD
False	0.0	83.6 %	32.7 %	61.1 %	68.7 %
	0.5	77.1 %	32.5 %	45.1 %	68.2 %
	0.95	76.1 %	32.4 %	24.3 %	71.9 %
True	0.0	88.2 %	34.0 %	68.0 %	69.3 %
	0.5	87.7 %	35.8 %	54.3 %	70.4 %
	0.95	86.8 %	36.1 %	30.9 %	75.4 %

3) Impact of CTC Regularization: To show the importance of the CTC regularization we evaluated the system with and without this regularization, the results are shown in table III. On both datasets, the CTC regularization has a significative impact on the results, especially on ILGDB. Because of the CTC loss which encourages the predictions to be consistent between frames, the precision and the earliness is particularly impacted: +1.6 % of precision for MTGSetB and +8.8 % for ILGDB. However, it has a negative impact on the earliness, +0.4 % and +2.7 %, the gestures are detected slightly later. We can deduce that the network preferred to postpone more its detection to avoid detection instability, which is consistent with an application scenario.

A example of detection of the network trained with and without the CTC regularization is shown in figure 6. We can see on the last gesture that the detection switch between three gestures, this is the kind of behaviour which is hardly penalized by the CTC loss.

TABLE III
IMPACT OF THE CTC REGULARIZATION, UNTRIMMED CONTEXT, BOD METRIC WITH $canCorrect = False, \Delta = 0.25$

Dataset	Variant	Precision	Recall	FMeasure	NDtoD
MTGSetB	CTC	**70.7 %**	**87.0 %**	**78.0 %**	32.7 %
	No CTC	69.1 %	85.2 %	76.3 %	**32.3 %**
ILGDB	CTC	**56.4 %**	**55.6 %**	**56.0 %**	69.2 %
	No CTC	47.6 %	54.2 %	50.7 %	**66.5 %**

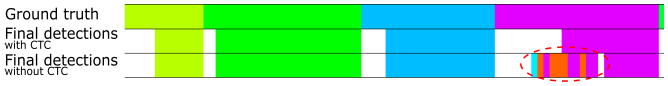


Fig. 6. Example of detections on MTGSetB with the network trained with CTC (second line) and without CTC (third line). We see that the last gesture is detected differently with the two versions, the final detection is more unstable without CTC.

4) Qualitative Results: Figure 7 (top) shows an example of a sequence of MTGSetB with the predictions and detections of the network. For the first gesture, we see that the network rejects predictions (i.e., confidence below 0.5) until the beginning of the second stroke of the "X" gesture to be sure not to confuse with the "W" gesture which is also contained in the dataset. For the next two gestures, it also waits until the common parts with other gestures are passed. We observe a similar behaviour on ILGDB (figure 7, bottom), but due to the difficulty of this dataset, it has difficulties to keep a consistent confidence score, which can produce false positive and missed detection.

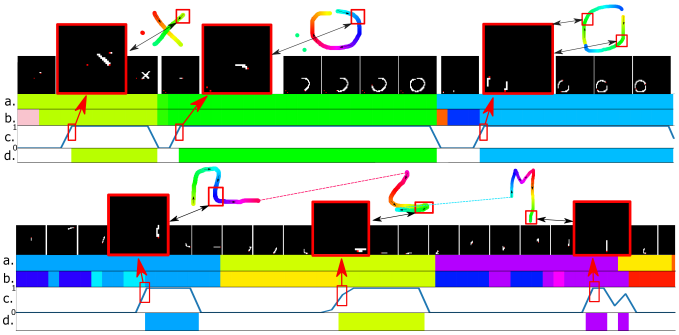


Fig. 7. Example of detections on MTGSetB (top) and ILGDB (bottom). a. Ground truth, b. Class predictions, c. Confidence, d. Final detections. The network waits until decisive instants to get a high confidence. Note that some intermediate frames have been removed for the visibility.

V. CONCLUSION

We presented an approach to address the challenging task of early recognition of untrimmed gestures. First, new representation strategies are designed to consider sequences of multi-stroke or mono-stroke gestures. We proposed to regularize the spatio-temporal CNN using the CTC loss to bring predictions stability. Moreover, we propose a new evaluation protocol with a new metric, and two artificial datasets. Our method obtained superior results when compared with other approaches in trimmed context. Good results are obtained in untrimmed context, which will be a strong baseline for future works.

Applying the CTC loss can be an open door to weakly-supervised training, using only the sequence-level annotation, this will be explored in future works. We will also address the task of early action recognition of a full 3D human body.

ACKNOWLEDGMENT

This study is funded by the ANR within the framework of the PIA EUR DIGISPORT project (ANR-18-EURE-0022).

REFERENCES

[1] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and 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)*, June 2016.

[2] M. Weber, M. Liwicki, D. Stricker, C. Scholzel, and S. Uchida, "Lstm-based early recognition of motion patterns," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 3552–3557.

[3] V. Bloom, V. Argyriou, and D. Makris, "Linear latent low dimensional space for online early action recognition and prediction," *Pattern Recognition*, vol. 72, pp. 532–547, 2017.

[4] S. Y. Boulahia, E. Anquetil, F. Multon, and 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, Jun. 2018, pp. 1–8.

[5] S. Uchida and K. Amamoto, "Early recognition of sequential patterns by classifier combination," in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

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

[7] W. Mocaër, E. Anquetil, and R. Kulpa, "Online spatio-temporal 3d convolutional neural network for early recognition of handwritten gestures," in *Document Analysis and Recognition – ICDAR 2021*, J. Lladós, D. Lopresti, and S. Uchida, Eds. Cham: Springer International Publishing, 2021, pp. 221–236.

[8] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[9] M. Yamagata, H. Hayashi, and S. Uchida, "Handwriting prediction considering inter-class bifurcation structures," in *17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 103–108.

[10] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe, "Early recognition and prediction of gestures," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, 2006, pp. 560–563.

[11] M. Kawashima, A. Shimada, H. Nagahara, and R. Taniguchi, "Adaptive template method for early recognition of gestures," in *17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2011, pp. 1–6.

[12] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3551–3560.

[13] R. Wang, J. Liu, Q. Ke, D. Peng, and Y. Lei, "Dear-net: Learning diversities for skeleton-based early action recognition," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[14] S. Y. Boulahia, E. Anquetil, F. Multon, and R. Kulpa, "Cudi3d: Curvilinear displacement based approach for online 3d action detection," *Computer Vision and Image Understanding*, vol. 174, pp. 57 – 69, 2018.

[15] H. J. Escalante, E. F. Morales, and L. E. Sucar, "A naïve bayes baseline for early gesture recognition," *Pattern Recognition Letters*, vol. 73, pp. 91 – 99, 2016.

[16] J. Liu, A. Shahroudy, G. Wang, L. Duan, and A. C. Kot, "Skeleton-based online action prediction using scale selection network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1453–1467, 2020.

[17] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, 2016.

[18] Y. Geifman and R. El-Yaniv, "Selectivenet: A deep neural network with an integrated reject option," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2151–2159.

[19] N. Renau-Ferrer, P. Li, A. Delaye, and E. Anquetil, "The ilgdb database of realistic pen-based gestural commands," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 3741–3744.

[20] Z. Chen, E. Anquetil, H. Mouchère, and C. Viard-Gaudin, "Recognize multi-touch gestures by graph modeling and matching," in *17th Biennial Conference of the International Graphonomics Society*, ser. Drawing, Handwriting Processing Analysis: New Advances and Challenges. Pointe-a-Pitre, Guadeloupe: International Graphonomics Society (IGS) and Université des Antilles (UA), Jun. 2015.