Online Handwriting Trajectory Reconstruction from Kinematic Sensors using Temporal Convolutional Network

Wassim Swaileh, Florent Imbert, Yann Soullard, Romain Tavenard, Eric Anquetil

Online Handwriting Trajectory Reconstruction from Kinematic Sensors using Temporal Convolutional Network

Wassim Swaileh[1*†], Florent Imbert[1*†], Yann Soullard[1,2], Romain Tavenard[2] and Eric Anquetil[1]

[1]IRISA, University of Rennes, INSA Rennes, France.
[2]LETG UMR 6554 CNRS, Université Rennes 2, 35043 Rennes, France.

*Corresponding author(s). E-mail(s): wassim.swaileh@irisa.fr; florent.imbert@irisa.fr;
†These authors contributed equally to this work.

Abstract

Handwriting with digital pens is a common way to facilitate human-computer interaction through the use of Online Handwriting (OH) trajectory reconstruction. In this work, we focus on a digital pen equipped with sensors from which one wants to reconstruct the OH trajectory. Such a pen allows to write on any surface and to get the digital trace, which can help learning to write, by writing on paper, and can be useful for many other applications such as collaborative meetings, etc. In this paper, we introduce a novel processing pipeline that maps the sensor signals of the pen to the corresponding OH trajectory. Notably, in order to tackle the difference of sampling rates between the pen and the tablet (which provides ground truth information), our preprocessing pipeline relies on Dynamic Time Warping to align the signals. We introduce a dedicated neural network architecture, inspired by a Temporal Convolutional Network, to reconstruct the online trajectory from the pen sensor signals. Finally, we also present a new benchmark dataset on which our method is evaluated both qualitatively and quantitatively, showing a notable improvement over its most notable competitor.

Keywords: Online Handwriting, Trajectory Reconstruction, Digital Pen, Temporal Convolutional Neural Network, Inertial Measurement Units

1 Introduction

Digital devices can help pupils and teachers in the learning process by promoting active learning techniques and providing immediate feedbacks [26]. The e-learning literature shows that computer-based analysis of handwriting can be really accurate, sensitive, and reliable to produce relevant and consistent feedbacks for correction or guidance. Several pen-based tablet applications have been designed in order to give immediate and personalized feedback to children [13]. Moreover, children still need to learn to write on paper

to acquire different sensations because today, it is still by far the most widely used handwriting medium.

As an answer to that need, digital pens have been designed to allow for handwriting on paper while capturing the handwriting gesture. Here, we focus on such kinds of stylus with the goal to reconstructing the digital handwriting trajectory of the pen. The digital pen that we use in this work is the Digipen stylus developed by STABILO, which is equipped of kinematic sensors to track the pen movements.

Nowadays, a wide range of applications in the domain of remote sensing and tracking systems benefits from recent improvements in deep neural network architectures. Tracking systems are commonly based on Inertial Measurement Unit (IMU) due to the low cost of these sensors. However, IMU-based sensors are quite imprecise due to the poor quality of the IMU signals.

IMU sensors are utilized in other fields to recognize pre-defined movements [15, 30] or recreate pedestrian trajectories [6]. More closely related to this work, a wide range of studies have tackled the OH recognition task, which can be successfully accomplished despite the noisy nature of IMU sensor signals.

In this work, we focus on the more challenging task of trajectory reconstruction. Regarding OH recognition, there is one label for a global shape of handwriting. The model is trained to extract global displacement features to produce a prediction. This is in contrast with the OH reconstruction where there is one label per time frame. Thus, the model is trained to extract local displacement features based on a more or less local view to produce position predictions. In addition, hovering parts, i.e. when the pen is in air, have to be modeled in order to get a good position of the next touching stroke. This must be done without knowledge about the labeling, as there is no pen traces related to those parts. To the best of our knowledge, the recent work proposed by Wehbi et al. [29] is the second attempt to reconstruct OH trajectories from a digital stylus using deep neural networks. It generalizes the works from Ott et al. [21] to multiple writers. Older works explore more traditional approaches for this task such as Markovian models [24] or movement acceleration inference techniques [3].

In terms of evaluation, some of the earlier works relied on a qualitative assessment of the reconstructed trajectories [18]. In other works, recognition performance is used as a proxy to evaluate reconstruction [10, 29]. Sometimes, additional reconstruction-only metrics such as the Root Mean Squared Error (RMSE) or the Dynamic Time Warping (DTW) are used [4].

This work presents a novel handwriting trajectory reconstruction pipeline from IMU sensors. Our main contributions are:

- preprocessing including an alignment strategy between input and ground-truth time series for training, as well as a learning pipeline based on touching strokes,
- a neural network architecture inspired by Temporal Convolutional Networks (TCN) to cope with noisy signals from the inertial sensors,
- a database that will serve as a benchmark to help advance research,
- an evaluation protocol based on the Fréchet distance to assess the quality of the reconstructed trajectories.

Both training and testing phases are described in details to allow the reproducibility of experiments.

The rest of the paper is organized as follows, related works are presented in the following section. Section 3 introduces the data acquisition protocol and challenges of this task. The proposed method is described in section 4. The experimental results are presented and discussed in section 5 before the conclusion and perspectives.

2 Related works

While the field of digital devices for note-taking, drawing, or handwriting learning is growing quickly, most of the systems use a screen for digital handwriting acquisition. Only few of them use a stylus that integrates motion tracking systems in order to reconstruct the handwriting trajectories. To the next, we first present related works on handwriting trajectory reconstruction, based on various devices. Then, we focus on the use of IMU sensors for handwriting recognition which is a task that have been widely studied.

2.1 Handwriting trajectory reconstruction

There are several types of systems dedicated to acquisition of digital handwriting. The first one is based on screens with special pen-based tablet. Tablet manufacturers, based on different Wacom technology, have developed their devices – such as Samsung Spen, Apple Pencil and Microsoft Stylet Surface Pen – to write on a tablet and enable to have a digital OH. These pen based tablets provide a precise track, but they have to be used on a specific device.

Another approach is to allow writing on paper, which is more natural, easier and more ergonomic. To do that, pens are equipped with specific tools to capture the handwriting gestures. One way is to embed a camera in the stylus, as for the Anoto. Disadvantages are that these stylus are expensive and that they have to be used on special papers.

A second way is to equip the stylus with IMU sensors. This solution is surface free and low cost and one can write on tablet, paper, or on a board. However, in contrast to a pen based tablet, where one obtains absolute coordinates of the online

trajectory, a signal from IMU sensors is only composed of relative pen displacements and it can be very noisy.

Based on the previous statement, we can distinguish between three categories of OH trajectory reconstruction tasks: i) trajectory reconstruction from offline handwriting image [4]; ii) reconstruction from a pen-tip optical tracking system [21]; iii) trajectory reconstruction from IMU signals [3, 21, 29]. In practice, only a limited number of works have focused on OH trajectory reconstruction from IMU signals. The works of Wehbi et al., [29] (multi-writer approach) and Ott et al, [21] (mono-writer non-generic approach) used deep learning techniques to reconstruct the OH trajectories from IMU signals as a step towards the OH recognition task. Earlier works such as [23] used more traditional approaches (Hidden Markov models [24], pen-tip acceleration and angular velocity inference to multi-level writing plans [3]) for the same task.

Bu et al. [3] introduces an IMU system based on accelerometer and gyroscope data to reconstruct handwriting. The method infers the tip trace displacements (captured by the IMU) into several planes of OH trajectories using principal component analysis (PCA) and a set of empirically chosen parameters. This method works in optimal conditions of use but remains sensitive to rotations and translations of the pen.

Ott et al. [21] rely on multi-task learning for joint classification and trajectory reconstruction. They show that a joint learning improved both the OH recognition and the trajectory reconstruction when using a suitable combination of loss functions. However, the proposed approach cannot be generalized since the proposed architecture is limited to produce trajectories of fixed length (100 points). Furthermore, the training and test sets were collected over a single writer.

Liu et al. [14] focus on magnetic signals but they are sensitive to outer magnetic field. In [24], the authors use low cost IMU from a smartphone to recognize characters and perform trajectory reconstruction. Linear Discriminant Analysis (LDA) is used to detect movements which are estimated mathematically.

Recently, Wehbi et al. [29] have proposed an approach that deals with the Stabilo Digipen, on its earlier version 5.0. They create a first dataset by using the Digipen on a tablet via a Stabilo mobile application. This allows to acquire simultaneously input signals from the pen and OH trajectories from a tablet.

The gap between sampling rates coming from the pen and the tablet makes mandatory to align the two time series to get one label for each time frame. It is done through a preprocessing that will be discuss more in details in Section 3.3. To do so, Wehbi et al. [29] linearly interpolate the OH trajectory signal from the tablet. However, the linear interpolation may re-distribute the points equally

in time, which result in the loss of the online writing dynamics of speeds and acceleration.

The authors observe that the linear interpolation on the whole output signal leads to erroneous alignments due to the variability of the transmission time delay. By applying the linear interpolation on strokes rather than on the whole label, better alignments are obtained at the cost of dropping out from the dataset any sample whose input and output signals have unequal numbers of strokes. The number of strokes is deduced from the force sensor signal (pen side) and the pressure signal (tablet side). Then, they use a Convolutional Neural Network (CNN) made of 3 convolutional layers with batch normalization layers in between to reconstruct the handwriting trajectory.

2.2 Using IMU sensors for handwriting recognition

As discussed before, only few works perform handwriting reconstruction from IMU sensors. On the other hand, the task of OH recognition from IMU sensor data has been tackled through the use of: i) the pen-tip trajectory signal given by a touchscreen device [16]; ii) signals coming from IMU sensors [3, 19, 21, 28, 29].

Recently, [19] introduced a benchmark study that compares several neural networks architectures trained to recognize characters, symbols, words and equations from IMU signals. A CNN/BLSTM architecture proposed in [22] obtains the best results. Other works intend to address both OH trajectory reconstruction and character recognition. For instance, Wehbi et al. trained two neural networks in succession: first, a model for reconstructing the OH trajectories from IMU signals; second, a model for character recognition based on the reconstructed trajectories. They assess the quality of the reconstruction based on the character recognition rate, but a good recognition rate may not reflect the quality of the reconstructed trajectory. A multitask learning is proposed by Ott et al. [21] using a CNN-LSTM network. They show that both the OH trajectory reconstruction and the character classification benefit from a joint learning.

As part of the UbiComp 2021 Challenge on OH recognition, Wegmeth et al. [27] trained a CNN/BLSTM to recognize mathematical expressions written with the Stabilo's Digipen. Their approach is based on a boundary feature extractor follow by a character classifier. As a consequence, the recognition performance depends on the quality of the label boundary splitting. Other studies minimize the link bandwidth between the Digipen and the remote device (e.g. tablet) [12] or explore domain adaptation [11, 20] and explainability [1] in the context of OH recognition from the Digipen.

3 Data acquisition

In this section, we present the protocol applied for collecting a new benchmark IRISA-KIHT dataset. We detail the data acquisition process and the protocol to generate the ground truth. For this, Stabilo [1] created the Digipen, a pen ball digital pen with a Wacom tip instead of an ink ball. Several versions of such digital pen exist, such as Digipen kids version 5.0 that works at 100Hz sampling rates and Digipen basic version 6.0 that works at 400Hz (the one used in this work).

3.1 Acquisition tools description

To create the training and test datasets of the handwriting trajectory reconstruction task, we use three equipments and tools: (i) the Digipen; (ii) a Wacom tablet operated with android OS; (iii) the Stabilo's application.

The Digipen embeds the following IMU sensors (Figure 1): a gyroscope, a front and a rear accelerometer, a magnetometer, and a force sensor. Each of these sensors has its own internal clocking system and provides temporal reading values that describe the relative pen movement in 3-axes channels (x, y and z), except the force sensor which has only one channel. The readings of the sensors are buffered before being sent to the tablet via Bluetooth connection. This makes 13 multi-variate time series for every dataset sample.

The tablet choice is important, as the model, the screen size, and the sampling rate must be considered to get consistent OH signals as ground truth. Indeed, each tablet has its own sampling rate so, only one model of tablet (Samsumg Galaxy S7 FE) has been kept to obtain homogeneous ground truth. The tablet's base sampling rate is 370Hz which is close to the Digipen one. Note that sampling rate can decrease to 60Hz depending on the writing.

Stabilo has developed a dedicated mobile application to record (i) the handwriting trajectories using the Digipen integrating Wacom tip, (ii) the corresponding sensor signals from the Digipen. The application allows to calibrate the Digipen sensors' signals and setup the sampling rate of the Digipen sensors.

3.2 Data acquisition protocol

The recording process (Figure 2(a)) starts by selecting one set of predefined scripts that will be written on the tablet surface using the Digipen. One set consists of 34 samples that have to be written one at a time during a single recording session. It is composed of five groups: 15 characters, 10 words, 5 equations, 2 shapes and 2 word

[1] https://stabilodigital.com/

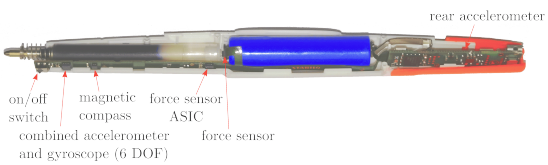

Fig. 1 © STABILO International Digipen's sensor location

groups. While recording, a user holds the pen's on/off switch up, which is a natural way to take the Digipen due to grips designed on the pen to naturally position the fingers properly.

3.3 Data acquisition challenges

To create the IRISA-KIHT dataset the acquisition process is quite challenging due to several difficulties. First, the stylus and the tablet have different sampling rates, so the selection of both of them is an important step. As discussed before, we select the Samsung S7 FE tablet and Digipen v6 for data collection. Second, an important degree of freedom comes from the pen orientation that directly impacts pen sensor values. The Digipen design includes grips to naturally position the fingers properly with the stylus on the same orientation. Another challenge concerns the recording of the hovering (pen up) movements. When the pen gets too far from the tablet, the tablet stops recording a trace. This results to parts of the data coming from the stylus that not match any ground truth from the tablet.

In addition, a drift in the accelerometer measurements is possible, and the kinematic signals are transmitted from the pen to the tablet in sets of six points through Bluetooth. Variable transmission time delay results in asynchronous timestamps for kinematic and tablet data.

4 Trajectory reconstruction pipeline

To reconstruct the handwriting trajectory from IMU signals, we present a complete pipeline which consists of training and test phases as illustrated in Figures 2, and 3. The training phase consists in: (i) a preprossessing process that produces cleaned and aligned input sensors signals and output handwriting trajectory signals (Figure 2(b) to (d)); (ii) a reconstruction training process, where a neural network model based on a Temporal Convolutional Network (TCN) architecture is trained on strokes using a dedicated loss function to predict the displacement vectors of the handwriting trajectory (Figure 2(e)); (iii) a post-processing to obtain the handwriting trajectory signal from the expected displacement vectors. (Figure 2(f)).

The test phase comprises the three principal steps of the training phase. However, preprocessing is reduced to only cleaning and normalizing the input signals (Figure 3(ii)). In addition, the prediction model (Figure 3(iii)) is applied on the whole input signal of the target sample, including touching and hovering strokes of the sample. Post-processing are the same as in the training phase followed by scaling and centering steps to compare the predicted trajectory with the ground truth.

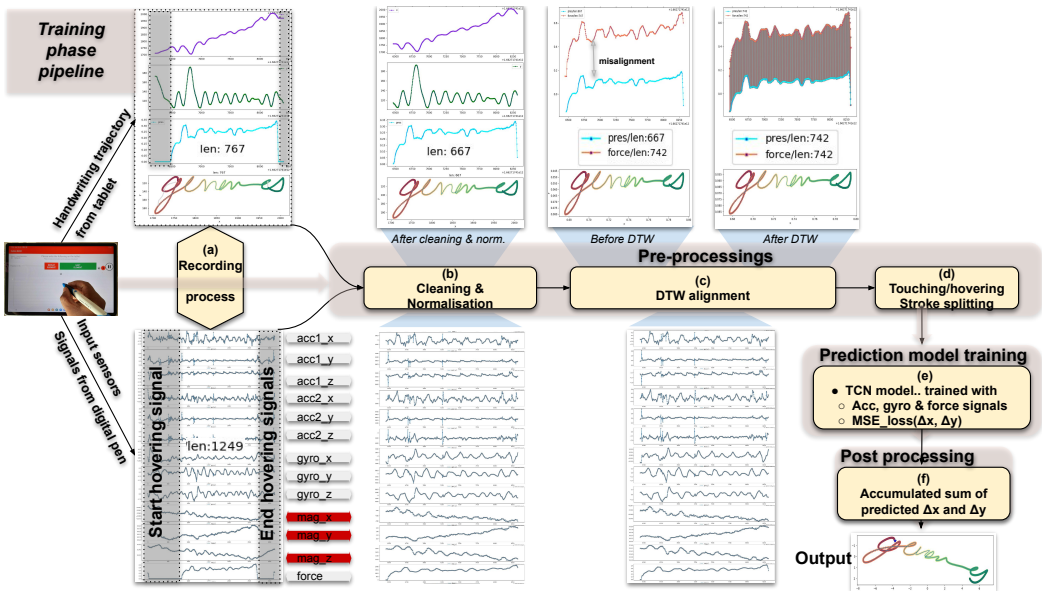

Fig. 2 Training processing steps of our proposed handwriting trajectory reconstruction pipeline.

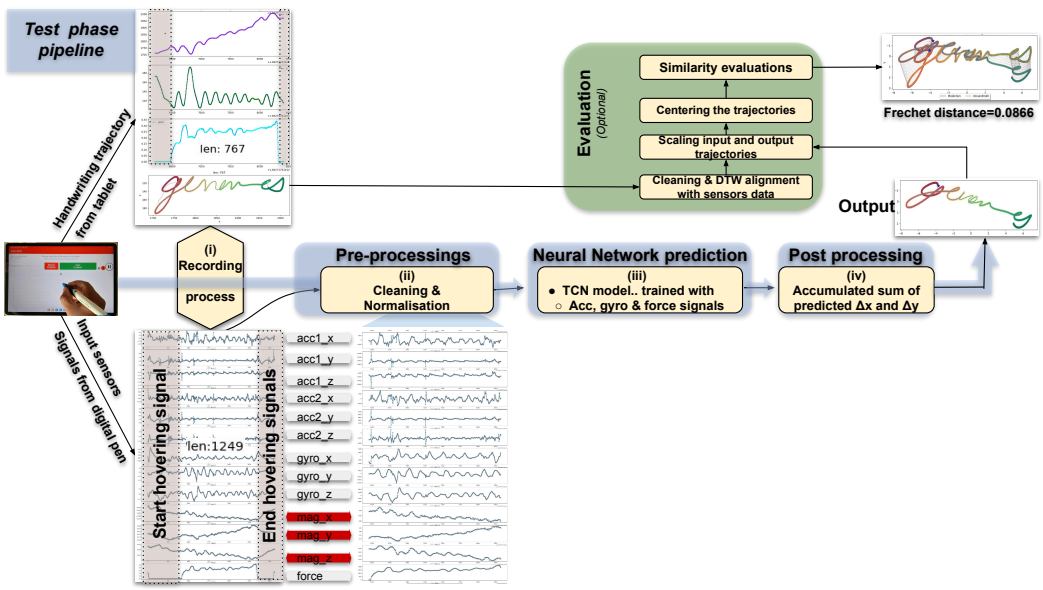

Fig. 3 Test processing steps together with post processing evaluation of the proposed handwriting trajectory reconstruction pipeline.

To the next, we describe the three main steps of our proposed pipeline: data processing, our neural network and post-processing. Every steps are discussed for training and test phases.

4.1 Data preprocessing

Data prepossessing are required at the training phase due to the misalignment of the pen and tablet signals and to the noise in signals. First, signals are divided into spans that correspond to

the written samples. One recalls that input signals captured from the Digipen are composed of 13 channels while signals from the tablet are made up of 3 channels (x, y and pressure). The timestamps of the Digipen and the tablet are added as an additional channel. Then, every sample is passed forward to the following process section 4.1.1.

4.1.1 Cleaning and normalization

As part of the training phase, irrelevant parts of the input signals are removed. Those parts are the start and end pen-up movements that does not refer to pen-up and pen-down actions to write the given script (Figure 2(b)). Since the input sensor signals represent the relative values of the acceleration, speed and orientation of the Digipen, it is reasonable to map it to the displacement vectors of the handwriting trajectory signal rather than to the real values of handwriting trajectory. Thus, displacement vectors (Δx, Δy) are computed from the (x, y) channels of the handwriting trajectory. In order to maintain the interoperability of the system between the different versions of Digipen, accelerometers, gyroscope, magnetometer and force signals are normalized by their maximum values (as reported by the manufacturer). In the test phase, the cleaning and normalization process concerns the sensors' signal only (Figure 3(ii)).

4.1.2 Input - Output DTW alignment

Due to the different sampling rates between the stylus and the tablet, an alignment process is crucial to get a mapping between the input and output time series for training. In addition, the sensor data are acquired in packets of 6 data points and the recorded timestamps are not equally spaced due the Bluetooth transmission time delay. Due to this mismatch, the recorded input and output signals have different lengths and they are not synchronized.

In order to respect the writing dynamics as much as possible, we propose an alignment approach based on the Dynamic Time Warping algorithm (DTW) (Figure 2(C)). The pen and tablet signals are recorded simultaneously. The transmission time delay is between 10 and 40 milliseconds in the Digipen version 6.0. Since the sampling rate of the sensor data is higher than the one of the tablet data, we need to up-sample the tablet data to match the sensors data length.

First, the average transmission time delay is subtracted to the tablet data in order to approximately synchronize it with sensors data. Then, we use the DTW algorithm (for more details see the appendix) to find an alignment path between the timestamps of the stylus and the tablet.

In practice, we observe that aligning the timestamps using DTW as we do is more effective than relying on the DTW alignment of force

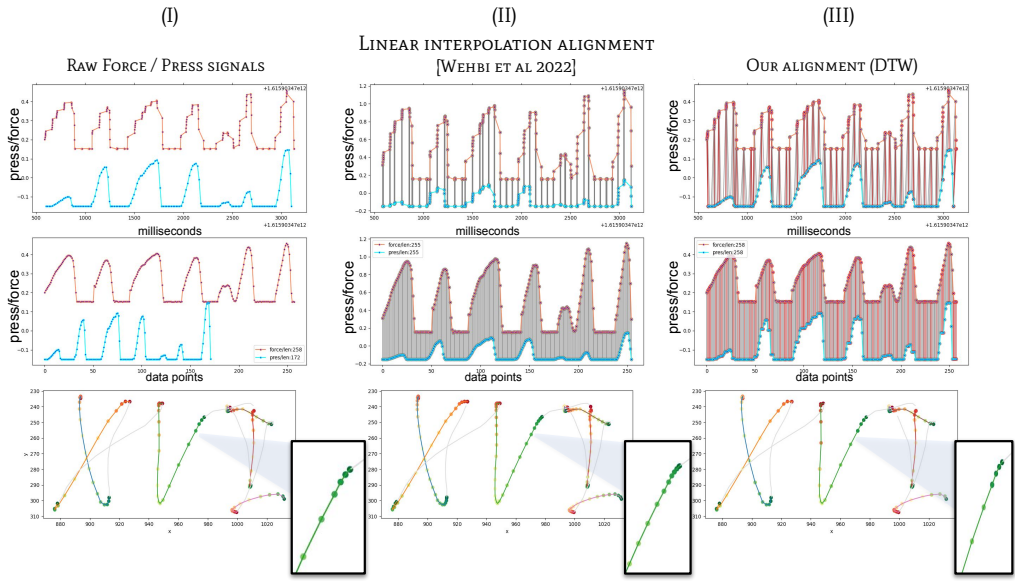

Fig. 4 Our proposed DTW based alignment compared to the linear interpolation based alignment proposed by [29]. The left side image represents the raw force / pressure (sensor/online) signals, the middle image shows the interpolation alignment result and the right side image shows the DTW based alignment results. The red lines connecting the two signals represent the duplicate points added to the pressure signal and the grey ones represent the one-to-one alignment.

and pressure data. Figure 4 shows a comparison between the alignment approach used in [29] based on linear interpolation and our proposed approach based on DTW. Alignments are presented on the force (pen sensor) signal against the pressure signal (from the tablet). The icon images of Figure 4 shows that the DTW alignment (Figure 4(III)) results in the same data distribution than the raw points while the linear interpolation one (Figure 4(II)) spaces the data points equally.

4.1.3 Splitting into strokes

Splitting into strokes only concerns the training phase of the proposed pipeline. It enables to train the neural network on the touching parts (strokes) only (Figure 2(d)). The idea is to use the force/pressure signals of the sensors and handwriting trajectory signals to determine the touching strokes of the training samples, knowing that we consider every touching stroke whose force/pressure values is greater than a predefined threshold, namely 0.01 for the force and 0 for the pressure.

We have the choice to either learn the neural network model on the entire training sample including the touching and hovering stroke, or to learn it on the touching strokes only. We will investigate it in the experimental part. However, we always test the model on the entire test samples (including the touching and hovering strokes).

4.2 Neural Network model

We design a neural network architecture to predict the displacement vectors of the handwriting

trajectory from the tablet, given a sensors signal of the Digipen (Figure 2(e)). The training and test steps of our proposed model are described in the following sub-sections.

4.2.1 Model architecture

In order to take into account the past and future states when mapping an input sequence toward an output signal, we propose to use a non-causal Temporal Convolutional neural Network (TCN) architecture as in [2]. The choice of a TCN architecture is supported by the great success of the TCN for sequence-to-sequence tasks with few training samples, e.g. for weather prediction [31], traffic prediction [5] and sound event localization and detection [8].

The CNN architecture proposed in [29] captures the spatial features that refer to the arrangement of data points of a sequence, and the relationship between them within the sequence. However, it lacks the hierarchical and distant dependencies that can be grasped with dilated convolutions and a deep architecture as in a TCN.

It was proved that TCN architecture is most suitable to extract relevant spatial and temporal features for a sequence of frames describing an action compared to an LSTM recurrent network [17]. Indeed, recurrent architectures are known to suffer from vanishing gradient problems in very long sequences, such as those produced by IMU sensors. Thus, TCN based systems outperforms their LSTM counterparts in different fields of application such as anomaly detection [7] or skeleton-based action recognition [17].

Our model receives 10 of 13 channels of the sensor data. Magnetometer data are neglected due to the interference of the tablet magnetic field with the signal. Figure 5 illustrates our network architecture. It is composed of TCN-like layers, a dense layer followed by a batch-normalization layer before the last dense layer that produces the output of the network (the two (Δx, Δy) channels). The TCN part consists of three stacked inner blocks of non-causal and dilated 1D-convolutions with kernel size of 5. The dilation rate applied to each block is increased with the depth of the network, from 1 to 4 and every inner layer is followed by a batch normalization layer. More details are given in appendix.

4.2.2 Model training and test

During the training phase, the model is trained to minimize the Mean Squared Error between the real and predicted displacement vectors of the handwriting trajectories. One uses the ADAM optimization, a batch size of 16 and a learning rate of $10e^{-3}$. As stated before, the 10 input channels of the sensor signal are the front and rear accelerators, gyroscope and force channels of the sensor signal, obtained after the cleaning and normalization process.

During the test phase, a cleaned and normalized signal is given as input of the model that predict the corresponding displacement vectors of the handwriting trajectory signal.

4.3 Post-processing

This process is the same in training and test phases. It consists in reconstructing the predicted handwriting trajectories by computing the accumulated values from the predicted displacement vectors (Figure 2(f), and Figure 3(iv)).

At test time, to evaluate the shape of the prediction, we normalize and center the predicted trajectory and the reference trajectory using the Procrustes analysis to fit a uniform scale.

5 Experiments and results

This section presents the experimental framework, especially the evaluation protocol and the metric used to evaluate our model. Then, we investigate the benefit of the successive parts of our approach. First, we experiment variants of networks to explore various receptive field sizes. Second, one compares our pipeline to a previous approach [29] and evaluates the benefit of using both our alignment method and our model. Finally, one explores the impact of training our model both on touching and hovering strokes compared to training it on touching strokes only, when the goal is always to predict both touching and hovering strokes.

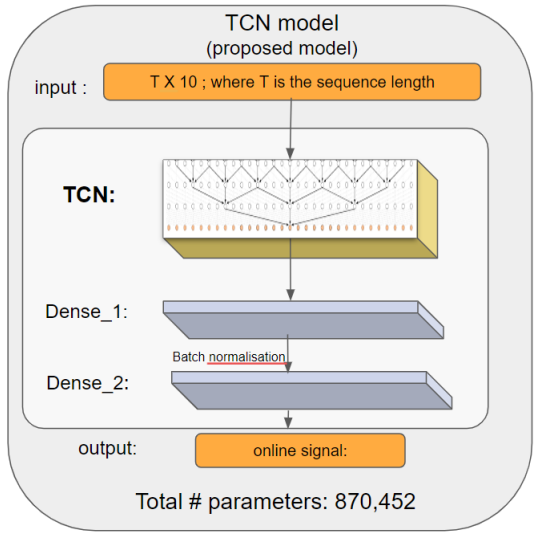

Fig. 5 Our TCN model for handwriting trajectory reconstruction.

5.1 Evaluation protocol and metric

Some works perform trajectory reconstruction as a step towards recognition. Thus, final recognition performance can be used as an evaluation criterion [29] and, as discussed before, it cannot reflect well the quality of the reconstruction. Here, the end goal is the trajectory reconstruction and dedicated metrics should be used to assess similarity between ground truth and reconstructed trajectories. Root Mean Squared Error (RMSE) is a natural candidate [21], but it can't handle temporal alignment. Dynamic Time Warping (DTW), allow this temporal alignment. Chen et al. [4] created the Adaptive Intersection over Union (AIOU) metric, which uses the stroke width and thickness as well as the length-independent

DTW to compare the trajectory-point alignment of two different handwriting trajectories of different lengths. In this work, we decide to rely on the Fréchet distance , whose formal definition is given in appendix section 6.4. This distance has benefits that we detail in the appendix section 6.4. The Fréchet distance measures the distance between curves, by taking into account the location and ordering of the points along the curves [9]. This allows to capture both local and global information accurately, and correlates well with qualitative assessment in practice.

In fact, due to the limited tracking capacity of the tablet over 15mm above the tablet screen, the handwriting signal is partially captured on hovering. For that, we only consider the touching parts (pen-down) for which we can compare a trajectory prediction to the handwriting trajectory from the tablet and compute the Fréchet distance.

We used the FAU-EINNS dataset, our IRISA-KIHT dataset, and a subset of it, IRISA-KIHT-S, to compare our processing pipeline to [29] IRISA-KIHT and IRISA-KIHT-S contain 38 and 30 writers' recordings, respectively. Each recording contains 34 characters, words, word groups, equations, and shapes. Section 6.2 of the appendix describes the datasets. The FAU-EINNS dataset consists of words written by 6 writers, and we take the samples of the user numbers 1, 2, 3, 5 and 6 for training (1774 samples) and testing on the 344 samples.

5.2 Receptive field effect

To design the TCN model architecture, we pay attention to the sampling rate of the input signal of sensors. It is obvious that the TCN based neural network may have the capacity to absorb the noise of such low quality and noisy signals, depending on the size of its receptive field.

In order to evaluate the receptive field effect of our model, we trained and evaluated the model with different sizes of receptive fields equal to 49, 85, 168 and 373 on both FAU-EINNS (100Hz) and IRISA-KIHT (400Hz) datasets.

Table 1 shows that the TCN model with the smallest receptive field size (49) performs better than the ones with greater receptive fields on FAU-EINNS (100Hz) dataset. On our IRISA-KIHT (400Hz) dataset, the TCN model with the greatest receptive field (373) generally outperforms the other sizes of receptive field as seen in Table 2. With a big receptive field, a large context can be exploited to make the prediction as well as the signal noise of long hovering can be absorbed, with the cost of higher number of learnable parameters. Table 3 illustrates the size of TCN models in terms of number of learnable parameters with regard to the receptive fields.

By comparing TCN-49 and TCN-373, during hovering strokes, the network may see the last and next touching points of two successive touching strokes, as illustrated in Table 2 with better

Table 1 Fréchet distance with standard deviation from TCN model with varied receptive fields on FAU-EINNS dataset

Type	TCN-49	TCN-85	TCN-169	TCN-373
Global	**0.077** ± 0.0356	0.0867 ± 0.0388	0.0893 ± 0.0426	0.0966 ± 0.0335

Table 2 Fréchet distance with standard deviation from TCN model with varied receptive fields on IRISA-KIHT dataset

Type	TCN-49	TCN-85	TCN-169	TCN-373
Global	0.2039 ± 0.1427	0.2415 ± 0.1676	0.2042 ± 0.1419	**0.1807** ± 0.1338
Characters	0.2758 ± 0.1709	0.3048 ± 0.2087	**0.236** ± 0.1901	0.2402 ± 0.1642
Words	0.1383 ± 0.0727	0.1716 ± 0.0886	0.1605 ± 0.0627	**0.1207** ± 0.0702
Equation	**0.1438** ± 0.0568	0.213 ± 0.0998	0.1964 ± 0.0639	0.1462 ± 0.0584
Shapes	0.281 ± 0.1071	0.3069 ± 0.1289	0.2758 ± 0.12	**0.2505** ± 0.1195
Word groups	0.1552 ± 0.0665	0.1971 ± 0.0775	0.2028 ± 0.0843	**0.1269** ± 0.0522

results on word groups. These two models are quite similar in terms of performance (Figure 6). As one of our next goal is to embed the model inside the Digipen, we decided to keep the TCN-49 model to get a good trade-off between the model performance and the number of parameters.

5.3 Comparison with related work

We compare the performance of our approach against the one of Wehbi et al. [29] on both FAU-EINNS dataset and our own dataset, called IRISA-KIHT. We also provide results on the subset of the latter called IRISA-KIHT-S which is made publicly available and can hence be used as benchmark for future works. We applied the same evaluation protocol on both models and processing pipelines. Due to the comparatively smaller size of IRISA-KIHT-S, a 3 fold cross validation is used when this dataset is at stake, as shown in Table 4. Results show that our pipeline (TCN-49 model and DTW based alignment) outperforms

Table 3 TCN models' size with various receptive fields

# Params	TCN-49	TCN-85	TCN-169	TCN-373
Total	467,452	528,452	870,452	894,452
Trainable	464,152	524,752	866,752	888,352

Table 4 Average Fréchet distance with standard deviation on IRISA-KIHT-S over 3-fold compared to [29]

	CNN model [29]	Our approach
Fold 1	0.4055 ± 0.2699	**0.2099** ± 0.1750
Fold 2	0.5188 ± 0.3926	**0.3122** ± 0.1731
Fold 3	0.4383 ± 0.3701	**0.1766** ± 0.1056
Mean	0.4542 ± 0.3442	**0.2329** ± 0.1512

Table 5 Average Fréchet distance with standard deviation of our pipeline compared to [29]

	CNN model [29]	Our approach
FAU-EINNS	0.2628 ± 0.1256	**0.077** ± 0.0356
IRISA-KIHT	0.4691 ± 0.2465	**0.2039** ± 0.1427
IRISA-KIHT-S	0.4542 ± 0.3442	**0.2329** ± 0.1512

Wehbi et al.[29] approach (CNN model and linear interpolation alignment) on every dataset, both quantitatively as seen in Table 5 on the average Fréchet distance and qualitatively as illustrated in Figure 7 (Wehbi et al. [29] appears on the second line of the figure while our approach appears on the bottom line). Despite the difference in size between IRISA-KIHT and IRISA-KIHT-S datasets, one can observe that reported performance is comparable in both datasets for both our method and its competitor (see table 5). This confirms that IRISA-KIHT-S is of sufficient size for the quantitative assessment of OH trajectory reconstruction from IMU sensor data. In the next section, we will evaluate each step of the processing chain.

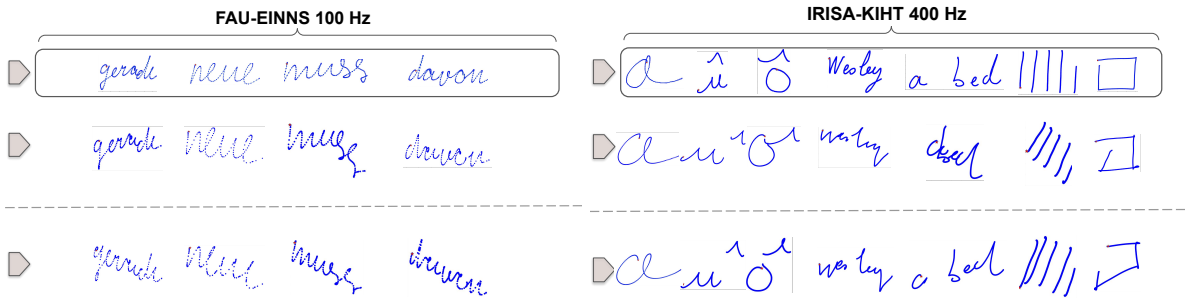

Fig. 6 Handwriting reconstructions with two TCN models with different receptive fields on the FAU-EINNS (left) and IRISA-KIHT (right) datasets. First line, ground-truth. Second, the 49-receptive-field TCN model. Third line: TCN model with 373 receptive field.

5.4 Alignment methods and models

In the following, we compare two effective processes of the handwriting trajectory reconstruction pipeline; (i) the input-output alignment process and (ii) the reconstruction model performance. We provide a cross validation scheme between the two alignment methods (linear interpolation and DTW based alignment methods) and the two reconstruction models (CNN and TCN based models) on the FAU-EINNS and IRISA-KIHT datasets. From Table 6, we observe that each step of our approach outperforms the counterpart of the Wehbi et al. [29] method on both datasets. The results show that our alignment is better whatever the model on all the datasets.

In contrast to the linear alignment, our alignment based on the DTW algorithm keeps the writing dynamic, which seems to be essential to reach quality trajectory reconstruction. Figure 7 shows examples of reconstruction trajectories using the CNN or TCN models, when using the linear interpolation and the DTW alignments on the two datasets. Moreover, our TCN model outperforms the CNN counterpart whatever the alignment method, which shows that the TCN model benefits from a larger receptive field and a deeper network to model more complex patterns and to be less sensitive to the noise.

5.5 Touching versus hovering trajectories reconstruction

Recalling the definition of the touching and hovering strokes, a touching stroke consists of data points that are recorded while touching the tablet screen. Hence, hovering stroke is recorded partially due to the limited capacity of the tablet screen to track the pen when it goes up to 15mm[2]. As a result, only the Digipen continue to provide sensor signals when the pen is up over 15mm. In conclusion, there is no ground truth for such hovering movements. To offset the effect of over hovering on the data preprocessing, the proposed DTW alignment duplicates the end points of the period when

[2]http://tennojim.xyz/article/wacom_intuos_pro_l_guide

Table 6 Model and alignment method comparison between [29] and ours on both datasets.

	CNN model [29]		Our TCN model-49	
	Linear interpolation alignment [29]	Our DTW alignment	Linear interpolation alignment [29]	Our DTW alignment
FAU-EINNS 100Hz	0.2628 ± 0.1256	0.0801 ± 0.0352	0.1211 ± 0.0639	**0.077** ± 0.0356
IRISA-KIHT 400Hz	0.4691 ± 0.1427	0.2095 ± 0.1255	0.3859 ± 0.2003	**0.2039** ± 0.1427

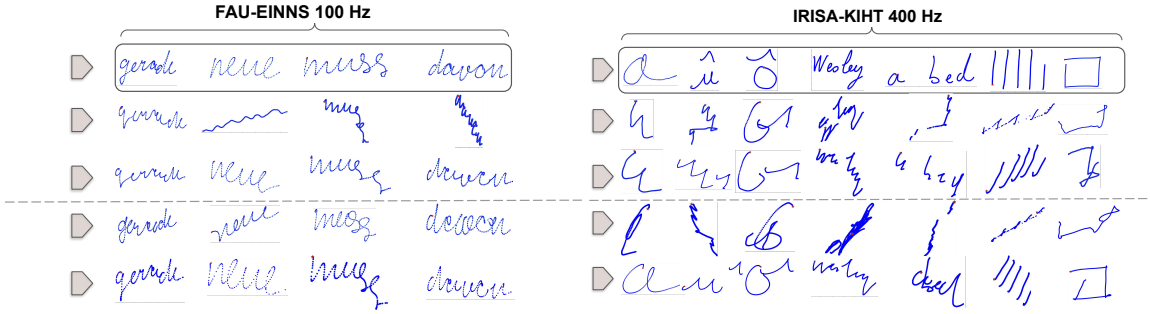

Fig. 7 Comparison with [29] approach with our one. First line, [29] CNN model and linear interpolation alignment results. Second line, CNN model and the introduced DTW alignment results. Third line, our TCN model and linear interpolation alignment results. Last line, our TCN model and the proposed DTW alignment results.

the pen is hovering. Figure 8 illustrates the effect of over hovering on the alignment process.

As illustrated in Figure 8, we distinguish two kinds of hoverings; (i) tracked hoverings, where the Digipen is still within the tracking field of the tablet. This kind of hovering is represented by the

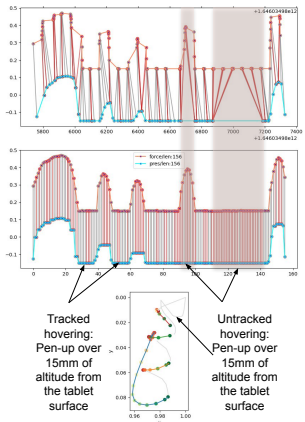

Fig. 8 Tracked hovering and untracked hovering effects on the alignment process

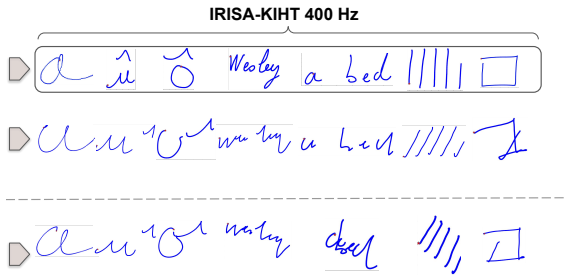

Fig. 9 TCN-49 model performance when training on hovering and touching data (middle) and touching data only (bottom). The ground truth is on first line.

Table 7 Training on touching strokes versus learning on touching & hovering strokes results

	Our TCN Model-49	
Type	Trained on touching & hovering strokes	Trained on touching strokes only
Global	0.2471 ± 0.1711	**0.2039** ± 0.1427
Characters	0.3587 ± 0.185	**0.2758** ± 0.1709
Words	0.1497 ± 0.0671	**0.1383** ± 0.0727
Equations	0.1483 ± 0.0666	**0.1438** ± 0.0568
Shapes	0.429 ± 0.2094	**0.281** ± 0.1071
Word groups	**0.1497** ± 0.0696	0.1552 ± 0.0665

flat doted lines of Figure 8. (ii) untracked hoverings, when the Digipen goes too far from the

tracking field of the tablet and consequently leading to lose the OH signal to be reconstructed by the model. At the same time, we observe a dual effect of missing the handwriting signal that can be correlated with the writer experience in writing. In fact, a fluent writer (adult) don't need to search where to start writing the next stroke of the same character or words. As a result, he don't need to raise the pen so high (outside the tablet' tracking field). However, for sentences, the user needs to space words and he may raise the pen higher while searching where to put the pen on the screen again.

To avoid the untracked hoverings in the training datasets, we used the touching data only to train our model. The idea is to train the model on touching strokes only and test on touching and hovering strokes as well. We compared our model when it is trained on touching only or on touching and hovering strokes.

Table 7 shows that the strategy of learning on touching strokes only is generally better than the one of learning on touching and hovering strokes upon the distance of Frechet, and this is also the case for characters, words, equations and shapes categories. Results on the sentence category are very close. We think this is due to the correlated effect of the untracked hovering and the pen-movement hesitations of the writer between the words of the sentence. However, by looking at Figure 9, we observe that the touching model reconstructs better touching strokes that appear in the form of single character (e.g. "a") or shape (e.g. "rectangle"). It is also better with double strokes characters (e.g. "û", "ô") where a single tracked hovering can be observed. Similarly, we observe that our model reconstructs quite good the touching parts of the word groups but it fails to find where to start the reconstruction of the next stroke. This may happen due to untracked hoverings (that represents movement hesitations about where to put the pen again on the screen) like in "a — bed" where there is a long untracked hovering between "a" and "bed".

6 Conclusion & perspectives

This work introduces a novel processing pipeline for reconstructing handwriting trajectories from kinematic sensor signals. Input signals come from a digital pen designed by the Stabilo company called Digipen. Our approach consists in training and testing phases, each one composed of three main processes; preprocessing, a neural network training/prediction process and post-processing. A new dataset has been collected using a mobile application, a tablet and a Digipen and we provide benchmark results on it. To tackle with the sampling rate differences between the Digipen and the tablet, as well as the synchronization problem, we introduced a DTW based alignment approach. We show through experiments on several datasets

that the TCN neural network architecture outperforms the CNN one proposed in Wehbi et al. [29] as well as the DTW based alignment approach that enhances the performance of both CNN and TCN models. We also show that the model predicts better the touching and tracked hovering parts, when the model is trained on the touching strokes only.

The major limitation of our approach is the modelling of the untracked / complex hovering trajectories that is still an open research problem. This constitutes a first important track for future works and could be solved using a semi-supervised or unsupervised deep learning framework. Moreover, in this paper, we only evaluated the shape of the reconstruction. We could extend our evaluation to the dynamics of the trajectory reconstruction such as speed, acceleration, etc.

Acknowledgments

This project is financed by the KIHT French-German bilateral ANR-21-FAI2-0007-01 project and these four partners, IRISA, KIT, Learn & Go and Stabilo. This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011013148)

References

[1] H. Azimi, S. Chang, J. Gold, et al. Improving accuracy and explainability of online handwriting recognition. *arXiv preprint arXiv:2209.09102*, 2022.

[2] S. Bai, J Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv*, 2018.

[3] Y. Bu, L. Xie, Y. Yin, et al. Handwriting-assistant: Reconstructing continuous strokes with millimeter-level accuracy via attachable inertial sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2021.

[4] Z. Chen, D. Yang, J. Liang, et al. Complex handwriting trajectory recovery: Evaluation metrics and algorithm. In *Proceedings of the Asian Conference on Computer Vision*, pages 1060–1076, 2022.

[5] R. Dai, S. Xu, Q. Gu, et al. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[6] S. Derrode, H. Li, L. Benyoussef, et al. Unsupervised pedestrian trajectory reconstruction from IMU sensors. In *TAIMA 2018: Traitement et Analyse de l'Information Méthodes et Applications*, Hammamet, Tunisia, 2018.

[7] S. Gopali, F. Abri, S. Siami-Namini, et al. A comparative study of detecting anomalies in time series data using lstm and tcn models.

arXiv preprint arXiv:2112.09293, 2021.

[8] K. Guirguis, C. Schorn, A. Guntoro, et al. Seld-tcn: Sound event localization & detection via temporal convolutional networks. In *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021.

[9] S. Har-Peled et al. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, 2002.

[10] H. Huang, D. Yang, G. Dai, et al. Agtgan: Unpaired image translation for photographic ancient character generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5456–5467, 2022.

[11] A. Klaß, S. M Lorenz, M. Lauer-Schmaltz, et al. Uncertainty-aware evaluation of time-series classification for online handwriting recognition with domain shift. *arXiv preprint arXiv:2206.08640*, 2022.

[12] F. Kreß, A. Serdyuk, T. Hotfilter, et al. Hardware-aware workload distribution for ai-based online handwriting recognition in a sensor pen. In *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4. IEEE, 2022.

[13] O. Krichen, S. Corbillé, É. Anquetil, et al. Combination of explicit segmentation with seq2seq recognition for fine analysis of children handwriting. *International Journal on Document Analysis and Recognition (IJDAR)*, 2022.

[14] Y. Liu, K. Huang, X. Song, et al. Maghacker: Eavesdropping on stylus pen writing via magnetic sensing from commodity mobile devices. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, New York, NY, USA, 2020. Association for Computing Machinery.

[15] J. McIntosh, A. Marzo, and M. Fraser. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2017.

[16] A. Mustafid, J. Younas, P. Lukowicz, et al. Iamonsense: Multi-level handwriting classification using spatio-temporal information. 2022.

[17] M. Nan, M. Trăscău, A M. Florea, et al. Comparison between recurrent networks and temporal convolutional networks approaches for skeleton-based action recognition. *Sensors*, 21(6):2051, 2021.

[18] H T. Nguyen, T. Nakamura, C T. Nguyen, et al. Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8320–8327. IEEE, 2021.

[19] F. Ott, D. Rügamer, L. Heublein, et al. Benchmarking online sequence-to-sequence

and character-based handwriting recognition from imu-enhanced pens. *arXiv preprint arXiv:2202.07036*, 2022.

[20] F. Ott, D. Rügamer, L. Heublein, et al. Domain adaptation for time-series classification to mitigate covariate shift. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5934–5943, 2022.

[21] F. Ott, D. Rügamer, L. Heublein, et al. Joint classification and trajectory regression of online handwriting using a multi-task learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 266–276, 2022.

[22] F. Ott, M. Wehbi, T. Hamann, et al. The onhw dataset: Online handwriting recognition from imu-enhanced ballpoint pens with machine learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–20, 2020.

[23] T-Y. Pan, C-H. Kuo, and M-C. Hu. A noise reduction method for imu and its application on handwriting trajectory reconstruction. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6, 2016.

[24] T-Y. Pan, C-H. Kuo, H-T. Liu, et al. Handwriting trajectory reconstruction using low-cost imu. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(3):261–270, 2018.

[25] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 1978.

[26] D. Simonnet, N. Girard, E. Anquetil, et al. Evaluation of children cursive handwritten words for e-education. *Pattern Recognition Letters*, 2019.

[27] L. Wegmeth, A. Hoelzemann, and K. Van Laerhoven. Detecting handwritten mathematical terms with sensor based data. *arXiv preprint arXiv:2109.05594*, 2021.

[28] M. Wehbi, T. Hamann, J. Barth, et al. Towards an imu-based pen online handwriting recognizer. In *International Conference on Document Analysis and Recognition*, pages 289–303. Springer, 2021.

[29] M. Wehbi, D. Luge, T. Hamann, et al. Surface-free multi-stroke trajectory reconstruction and word recognition using an imu-enhanced digital pen. *Sensors*, 2022.

[30] M. Wilhelm, D. Krakowczyk, and S. Albayrak. Perisense: Ring-based multi-finger gesture interaction utilizing capacitive proximity sensing. *Sensors*, 2020.

[31] J. Yan, L. Mu, L. Wang, et al. Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, 2020.

Appendix

6.1 DTW alignment

The DTW algorithm [25] is based on dynamic programming to assess the similarity between time series. For two multivariate time series $x \in \mathbb{R}^{T_x \times z}$, and $y \in \mathbb{R}^{T_y \times z}$ of equal feature dimensionality z and respective lengths T_x and T_y, the DTW is written as follows:

$$DTW(x, y) = \min_{\delta \in E(x,y)} \sum_{(i,j)\in\delta} d(x_i, y_j) \quad (1)$$

where $E(x, y)$ represents the set of all admissible alignments between x and y and d is a distance metric in $\mathbb{R}^z$. Commonly, the squared Euclidean distance $d(x_i, y_j) = \|x_i - y_j\|^2$ is used.

An alignment is a sequence of pairs of timestamps that is admissible if (i) it matches the first (and respectively the last) indices of time series x and y together, (ii) it is monotonically increasing, and (iii) it connects the two time series by matching at least one index of each series. Using dynamic programming, the admissible alignment paths can be computed according to the following recurrence formula:

$$DTW(x_{\to i}, y_{\to j}) = d(x_i, y_j) + \min \begin{cases} DTW(x_{\to i}, y_{\to j-1}) \\ DTW(x_{\to i-1}, y_{\to j}) \\ DTW(x_{\to i-1}, y_{\to j-1}) \end{cases}$$

where $x_{\to i}$ denotes time series x observed up to timestamp i.

The DTW method on timestamps links multiple sensor signal points to at least one tablet signal data point. In our case, we do not allow the association of multiple points of the tablet signal to one point of the sensor signal. In other words, in the previous recurrence formula the predecessor $DTW(x_{\to i}, y_{\to j-1})$ is removed. According to the resulting alignment path, the timestamps from the sensor signal are aligned with their corresponding timestamps from the tablet signal.

6.2 Datasets description

A group of 35 adult writers has contributed to the dataset acquisition. One recording is composed of 34 samples which are randomly selected scripts of characters, words, equations, shapes and word groups. To the next, this new dataset is called IRISA-KIHT. A writer can make several recordings that makes the IRISA-KIHT dataset to be writer imbalanced.

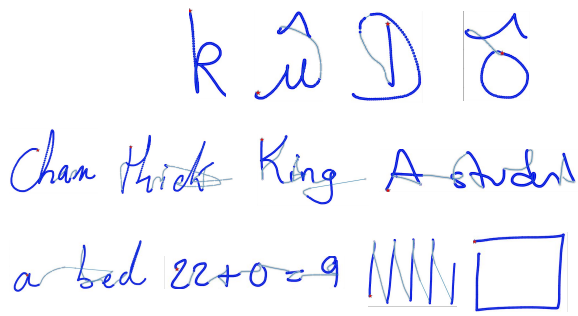

Fig. 10 Some examples of IRISA-KIHT data. pen-up (hovering) strokes are in gray and pen-down (touching) strokes are in blue.

Table 8 IRISA-KIHT and IRISA-KIHT-S datasets descriptions.

Datasets	Sets	# writers	# Recordings	# Samples	Characters	Words	Equations	Shapes	Word groups
IRISA-KIHT	Training	25	61 (mixed) + 53 (words)	3,770	915	2,306	305	122	122
	Test	10	10 (mixed)	340	150	100	50	20	20
	Total	35	124	4,110	1065	2,406	355	142	142
IRISA-KIHT-S (3-Folds Cross-validation)	Training	20	20 (mixed)	680	300	200	100	40	40
	Test	10	10 (mixed)	340	150	100	50	20	20
	Total	30	30	1020	450	300	150	60	60

To evaluate the generalization capacity of our proposed approach, we decided to train and test our model on different writers. The IRISA-KIHT dataset consists of 25 writers' recordings in the training set (3,770 samples) which represents around 71.5% of the writers' total number. The rest 10 writers' recordings never seen in training refer to the test set, representing 28.5% of writers (8.3% of samples). Figure 10 shows some samples of the IRISA-KIHT datasets.

The IRISA-KIHT-S dataset is a subset of the IRISA-KIHT dataset which is available on request[3]. This dataset is composed of 30 recordings and it is writer balanced as there is one recording per writer. Table 8 presents the IRISA-KIHT and IRISA-KIHT-S statistical description.

Every 34-sample recording session generates files from the data acquisition mobile app. The sensor signals file has 15 columns and N rows, where N is the number of IMU signals, timestamps, and sensor values. The table has 13 columns: milliseconds, accelerometer front (x, y, z), accelerometer rear (x, y, z), gyroscope (x, y, z), magnetometer (x, y, z), and force signals. Tablet signal files contain milliseconds, position coordinates (x, y, z), and pressure force signals. The transcription (labels) file contains labels and the start and stop time-stamps for every sample. Additional files concerning the sensor calibration and recording meta data are provided. The dataset website describes the format and meta data.

6.3 TCN architecture

A TCN (Temporal Convolutional Network) like layer consists of dilated, residual non-causal 1D convolutional layers with the same input and output lengths. The input tensor of our TCN implementation has the shape (batch_size=None, input_length=None, channels_num=13) and the output tensor has the shape (batch_size=None, output_length=None, channels_num=2). Since each TCN layer has the same input and output length, only the third dimension of the input and output tensors vary ("None" here denote the learning with batches of different size). The TCN

[3]Available free of charge for research community upon demand for research purposes only

layer consists of 4 blocks in TCN-49 and 3 blocks in TCN-373, with each block containing 100 filters followed by a RELU activation function and padded to the "same" input shape. A batch normalization layer was introduced between each pair of consecutive blocks. The performance of a TCN architecture based model depends on the size of the receptive field (RF) of the network. Since a TCN's receptive field depends on the network depth N as well as filter size k and dilation factor d, stabilization of deeper and larger TCNs becomes important [2]. The receptive field of our TCN based model is computed as :

$$RF = 1 + 2 \cdot (k-1) \cdot N \cdot \sum_{i} d_i \qquad (2)$$

The kernel size k is selected so that the receptive field covers enough context for predictions. For a regular convolution filter the dilation d is equal to 1. Using larger dilation enables an output at the top level to represent a wider range of inputs. In the equation, there is a multiplication by 2 because there are two 1D CNN layers in a single residual block of the TCN architecture. For the purposes of the trajectory reconstruction task, we set the stride equal to 1. For the TCN-49 we choose to set k=3, N=4 and d=1, 2 and for the TCN-373 k=3, N=3 and d=1, 2, 4, 8, 16.

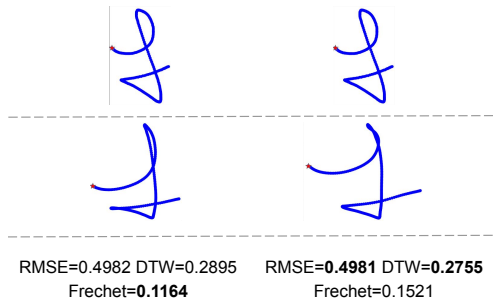

Fig. 11 Metrics comparison. On the top, the ground truth and on the bottom two predictions coming from different models.

6.4 Fréchetdistance

The formal definition of the Fréchet distance is: Let S be a metric space, d its distance function. A curve A in S is a continuous map from the unit interval into S, i.e. $A : [0,1] \to S$. A reparameterization α of $[0,1]$ is a continuous, non-decreasing, surjection $\alpha : [0,1] \to [0,1]$ Let A and B be two given curves in S. Then, the Fréchet distance between A and B is defined as the infimum over all reparameterizations α and β of $[0,1]$ of the maximum over all $t \in [0,1]$ of the distance in S between $A(\alpha(t))$ and $B(\beta(t))$. The Fréchet distance F is defined by the following equation:

$$F(A,B) = \inf_{\alpha,\beta} \max_{t \in [0,1]} \{d(A(\alpha(t)), B(\beta(t))\} \qquad (3)$$

Appropriate assessment Metrics are needed to evaluate trajectory reconstruction to give students useful feedback. Fréchet distance doesn't disagree with visual evaluation like DTW and RMSE. Figure 11 indicates that the left upper loop of the f is better reconstructed. Fréchet distance is needed to observe this.