Continuous pattern detection and recognition in stream
- a benchmark for online gesture recognition
Nehla Ghouaiel, Pierre-François Marteau, Marc Dupont

To cite this version:

HAL Id: hal-01764447
https://hal.science/hal-01764447
Submitted on 12 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Continuous pattern detection and recognition in stream
-A benchmark for on-line gesture recognition-

Nehla Ghouaiel*, Pierre-François Marteau*, Marc Dupont*†
*IRISA, Université de Bretagne Sud, Campus de Tohannic, Vannes, France
†Thales Optronique, 2 Avenue Gay Lussac, Elancourt, France

Abstract—Very few benchmark exists for assessing pattern detection and recognition in streams in general and for gesture processing in particular. We propose a dedicated benchmark based on the construction of isolated gestures and gesture sequences datasets. This benchmark is associated to a general assessment methodology for streaming processing which first consists in labelling the stream according to some heuristics (that can be optimized on training data) and then aligning the ground truth labelling with the predicted one. 6 pattern recognition models (including DTW, KDTW, HMM, HCRF and SVM) have been accordingly evaluated using this benchmark. It turns out that the regularized kernelized version of DTW measure (KDTW) associated to a SVM is quite efficient, comparatively to the other models, for detecting and recognizing continuous gestures in streams.

1. Introduction

This paper aims at presenting a benchmark dedicated to compare methods for gesture spotting and recognition in streams using low-cost sensors, such as the Microsoft Kinect or data gloves. Online human gesture recognition has a wide range of applications including video surveillance, human computer/robot interaction, sports video analysis, video retrieval and many other sensor-acquisition related application. Accurate spotting and recognition of human gestures in streams is still a quite challenging task, despite the research efforts in the past decade and many encouraging advances. We present in this paper a benchmark procedure that includes two datasets, an assessment methodology and a case study that involves benchmarking pattern detection and recognition models applicable to stream processing.

The remainder of this paper is organized as follows; Sec. 2 is dedicated to the state of the art of previous work in on-line gesture recognition. Sec. 3 describes the data set considered in the experiments. The technical approach is presented in Sec. 4. We detail carried out experiments and discuss obtained results in Sec. 5 before providing some conclusions and directions for future research.

2. Previous work in on-line gesture recognition

Stream processing requires (explicitly or implicitly) temporal segmentation of sequences of temporal data. For human motion it refers to the task of temporally cutting sequences into segments with different semantic meanings (gesture/action) which is an important step in gesture/action analysis and recognition. The existing approaches proposed to deal with this issue can be classified into two categories. The first category is about gesture recognition from color videos and hence is based on visual features. The second one rely instead on the features of motion data that describe the human-body-part movements.

On the one hand, for the video-based methods, it is a common practice to locate spatio-temporal interest points like STIP (1), and use the distributions of the local features like HOF (2) or HOG (3) to represent local spatio-temporal pattern. Li et al. (4) proposed the Bag of Visual Words (BoVW) model, which is used by many researchers for action recognition from color videos. BoVW model uses histograms as the features for gestures recognition. In order to count histograms, frames must be segmented first which makes BoVW model not suitable for online gesture recognition. Moreover, Song et al. (5) proposed a system for gesture recognition based on the combination of body and hand pose information. In their system, the body pose is estimated by using a multi-hypothesis Bayesian inference framework with a particle filter (6). A multi-class SVM classifier (7) is trained off-line based on HOG descriptors (3) extracted from manually-segmented images of hands, and is used to classify hand poses. Nevertheless, this system is not allowing non-segmented continuous time-series input.

On the other hand, several researchers claim that the movement of the human skeleton can be used for distinguishing different human gestures. In this context, the proposed approaches for gesture recognition are based on the features of motion data describing a specific human-body-part motion. Müller et al.(8) present an approach to label motion capture (MOCAP) data according to a given set of motion categories or classes, each specified by a set of motions. The presented method employs Motion Template (MT) to segment and label the motion data. A template is a generic gesture instance used to match with the data stream for a class of gestures. In their method, it is represented by a matrix averaging the training motions expressed by relational features. The gesture-level motion template approaches have a major weakness on dealing with intra-class variations which depend on the person performing the gesture. Consequently, this method is inefficient for dealing with real-time data
streams. Additionally, Wang et al. (9) proposed to learn one subset of human body joints for each action class. The subset joints are representative of one action compared to others. However, their approach is only applicable to the recognition of pre-segmented instances and cannot be used in online recognition of unsegmented data streams. Furthermore, Gong et al. (10) proposed an alignment algorithm for action segmentation. The proposed method called Kernelized Temporal Cut (KTC), is a temporal extension of Hilbert space embedding of distributions (11) with kernelized two-sample test (12) and was applied to sequentially estimate temporal cut points (change points) in human motion sequences. Since their approach is based on structure similarity between frames, it is only suitable for segmenting cyclic actions. Among others, Zhao et al. (13)(14) proposed a feature which they call Structured Streaming Skeletons (SSS) in order to represent inherent human motion characteristics. Each SSS feature vector consists of a number of attributes represented as distance values. Each value is a minimum DTW distance between all the scanned sub-sequences (ending at the current frame) and a template in the dictionary for the given pair of joints. Here a template is defined as a one-dimensional time series representing distance values of two joints of human body during the time of a gesture instance. The Jointly Sparse Coding (15) classifier was used to learn a gesture model from the extracted SSS feature vectors. Prediction is performed by a linear regression method that assigns each feature vector with a gesture label based on the learned gesture model.

In this paper, we tackle gesture recognition through the tracking of skeleton joint positions. Following (13)(14), to avoid segmenting streams beforehand, we develop an approach that simply requires the definition of the gesture vocabulary in the form of isolated patterns. These patterns are then used to detect and recognize gestures in streams. Stream segmentation is a by-product of the detection and recognition process.

3. Benchmark dataset

The datasets we have designed are complementary to the one developed by Microsoft Research (MSR Action Recognition Datasets) (16): they allow to explore the hand-shape and the upper body movement using 3D positions of skeletal joints.

To develop our benchmark data dedicated to upper body movement recognition, we have selected an excerpt of the Naval Air Training and Operating Procedures Standardization (NA TOPS) dataset presented in (5) which includes six of the twenty-four body-hand gestures used when handling aircraft on the deck of an aircraft carrier. The dataset includes automatically tracked 3D body postures and hand shapes using a Vicon high definition sensor. The body feature includes 3D joint velocities for left/right elbows and wrists, and is represented as a 12D input feature vector. The hand feature includes probability estimates of five predefined hand shapes: opened/closed-palm,thumb-up/down, and "no hand".

The simplified version of the NA TOPS database we have produced includes the gesture pair (i.e., #1 – #2, #3–#4, and #5–#6). As highlighted in (17), the gestures (#4, #5) and (#2, #3) are very similar and represent the intricacy of the entire set.

A good simplification of the complexity of the human body is to estimate only the position of joints since they apparently provide a sufficient representation of the human posture (18). In our database, each gesture is represented by a 24-dimensional feature vector, composed with the 3D-coordinates of hands (HandTips, thumbs and wrists) and arms (elbows). Figure 1 presents the location on the skeleton shape of the 3D-coordinates as blue (dark) dots.

The motion capture has been performed using a Kinect 2 sensor that produces a stream of 30 frames per second. Each frame consists in a 24 dimensional feature vector.

Isolated gesture dataset1. 20 subjects have been selected (15 male and 5 female) to perform in front of the sensor (at a three meters distance) the six selected NA TOPS gestures. Each subject repeated each gesture three times. Hence the isolated gesture dataset is composed of 300 gesture utterances that have been manually segmented to a fixed length of 51 frames (1.7 sec. duration).

Gesture sequence dataset1. The same 20 subjects also performed 3 times the previous mentioned gestures in a continual mode. The obtained dataset consists of 60 samples of motion data stream.

1. This dataset will be made available for the community at the earliest feasible opportunity

<table>
<thead>
<tr>
<th>#1: I have command</th>
<th>#2: All clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3: Not Clear</td>
<td>#4: Spread Wings</td>
</tr>
<tr>
<td>#5: Fold Wings</td>
<td>#6: Lock Wings</td>
</tr>
</tbody>
</table>

**TABLE 1.** The 6 selected gestures from the NATOPS data base; figures have been extracted from (5)
4. Methodology

4.1. Models

We hereinafter introduce the general classification models that we have used to learn and recognize isolated gestures and gestures in stream, namely first near neighbor with Dynamic time Warping (DTW) measure, first near neighbor with Kernelized DTW (KDTW), Hidden Markov Model (HMM), Hidden Conditional Random Fields (HCRF), Support Vector Machine (SVM) associated to DTW and KDTW.

4.1.1. HMM. An HMM (19) models a sequence of observations \( x = \{x_1, x_2, \ldots, x_T\} \) by assuming that there is an underlying sequence of states \( Y = \{y_t\}_{t=1}^T \) drawn from a finite state set \( S \). To model the joint distribution \( p(y, x) \) tractably, an HMM makes two independence assumptions. First, it assumes that each state depends only on its immediate predecessor. Second, an HMM assumes that each observation variable \( x_t \) depends only on the current state \( y_t \). With these assumptions, HMM can be specified using three probability distributions: first, the distribution \( p(y_1) \) over initial states; second, the transition distribution \( p(y_t|y_{t-1}) \); and finally, the observation distribution \( p(x_t|y_t) \). Therefore, the joint probability of a state sequence \( y \) and an observation sequence \( x \) factorizes as:

\[
p(y, x) = \prod_{t=1}^T p(y_t|y_{t-1})p(x_t|y_t). \tag{1}
\]

Each gesture is characterized by an HMM model and for each HMM, two procedures are required. In the training phase, the aim is to adjust the model parameters \( \lambda = (A, B, \pi) \) in order to maximize \( P(O|\lambda) \), where \( A \) is the state transition probability distribution, \( B \) is the observation symbol probability distribution and \( \pi \) is the initial state distribution. No tractable algorithm is known for solving this problem exactly, but a local maximum likelihood can be derived efficiently using the Baum-Welch algorithm (20). The Baum–Welch algorithm finds a local maximum for \( \theta^* = \arg \max_\theta P(Y|\theta) \) (i.e. the HMM parameters \( \theta \) that maximise the probability of the observation).

4.1.2. HCRF. The task performed by HCRF (21) is to predict the class \( y \) from the data \( x \), where \( y \) is an element of the set \( Y \) of possible gesture labels and \( x \) is the set of vectors of temporal observations \( x = \{x_1, x_2, \ldots, x_m\} \). Each local observation \( x_j \) is represented by a feature vector \( \Phi(x_j) \in \mathbb{R}^d \) where \( d \) is the dimensionality of the representation. The training set contains a set of labeled samples \( \{(x_i, y_i)\}, \text{ for } i = 1..n \) where \( y_i \in Y \) and \( x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\} \). For any \( x_i \) a vector of latent variables \( h = \{h_1, h_2, \ldots, h_m\} \) is assumed, providing the state sequence of the data. Each possible value for \( h_j \) is member of a finite set \( H \) of possible hidden states. HCRF is defined by a conditional probabilistic model:

\[
p(y, h|x, \theta) = \frac{\exp^\psi(y, h, x; \theta)}{\sum_{y', h} \exp^\psi(y', h, x; \theta)} \tag{2}
\]

Here \( \theta \) are the parameters of the model, and \( \Psi(y, h, x; \theta) \) is a potential function parameterized by \( \theta \). The function \( P(y|x, \theta) \) is defined by a summation over the \( h \) variables applied to the precedent equation.

\[
p(y|x, \theta) = \sum_{h} p(y, h|x, \theta) = \frac{\exp^\psi(y, h, x; \theta)}{\sum_{y', h} \exp^\psi(y', h, x; \theta)} \tag{3}
\]

Given a new test example \( x \) and parameter values \( \theta^* \) induced from the training set, the label for the example is taken to be \( \arg \max_{y \in Y} P(y|x, \theta^*) \). In the training phase, the following objective function is used for the estimation of parameter values \( \theta \):

\[
L(\theta) = \sum_i \log P(y_i|x_i, \theta) - \frac{1}{2\sigma^2} ||\theta||^2 \tag{4}
\]

where the first term in (4) is the log-likelihood of the data. The second term is the log of a Gaussian prior with variance \( \sigma^2 \). Under this criterion gradient ascent can be used to search for the optimal parameter values \( \theta^* = \arg \max_\theta L(\theta) \).

4.1.3. Support Vector Machine and time elastic kernels. The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser, Guyon, and Vapnik (22). An SVM classifies data by finding the best hyperplane, meaning the largest margin, that separates all data points belonging to the first class from those belonging to the other class. Hence, a Support Vector Machine seeks a decision function \( f \) which is defined in the space spanned by the kernel basis functions \( K(x, x_i) \) of the support vectors \( x_i \):

\[
y = f(x) = \sum_{i=1}^n w_i \ast K(x, x_i) + b. \tag{5}
\]

In 1995, to cope with non-separable cases, (23) suggested a modified maximum margin idea that allows for mislabeled examples. If there exists no hyperplane that can split the "positive" and "negative" examples, the "Soft Margin"
method will choose a hyperplane that splits the examples 
as properly as possible, while still maximizing the distance 
to the nearest proper split examples. By solving for the 
Lagrangian dual of the optimization equivalent problem, 
one obtains the simplified problem
\[
\max L(\alpha_1 \ldots \alpha_n) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \alpha_i K(x_i, x_j) y_j \alpha_j
\]
subject to \(\sum_{i=1}^{n} \alpha_i y_i = 0\), and \(0 \leq \alpha_i \leq C\) for all \(i\).

Where \(C\) is a penalty cost associated to the 
so-called slack variables. Since the dual minimization problem is a 
quadratic function of the \(\alpha_i\) subject to linear constraints, 
it is efficiently solvable by quadratic programming 
methods. In this paper, the kernel function 
\(K(\cdot, \cdot)\) required and the final kernel takes the form
\(K_{\text{rdtw}}(\cdot, \cdot) = e^{-\alpha \cdot \text{max}(K_{\text{rdtw}}(\cdot, \cdot)) / \sigma}\), 
with
\[
\begin{align*}
\alpha &= 1 / \log(\max(K_{\text{rdtw}}(\cdot, \cdot)) / \min(K_{\text{rdtw}}(\cdot, \cdot))) \quad \text{and} \\
\beta &= \exp(-\alpha \cdot \log(\min(K_{\text{rdtw}}(\cdot, \cdot))))
\end{align*}
\]
where \(\min, \max\) are taken over all the training data pairs.

5. Experiment

5.1. Assessing pattern recognition in stream

All recognition models considered in this paper provide 
as output a scoring file that contains for each processed 
time \(t\) a vector of scores (either a distance, a probability 
or any similarity or dissimilarity value) \(< S_i(t) >_{i=1..n_C}\), 
where \(n_C\) is the number of categories and \(S_i(t)\) is the 
score for the \(i\)th category at time \(t\). Two meta parameters 
are defined to labelled the stream from this vector or 
scores: \(S_0\), the similarity threshold value and the minimal 
duration value \(T_0\). Given these two thresholds, and supposed 
the score value has to be maximized, the labelling 
algorithm (LA) consists in outputting label \(C\) if and only 
if \(S_i(t) > S_j(t) \forall j \neq i\) and for \(t \in [t_n, t_{n+1}]\) such that 
\(t_{n+1} - t_n > T_0\), as depicted in Figure 2 for a binary 
classification problem.

The alignment of the ground truth labels with the predicted 
labelled stream \(< S_i(t) >_{i=1..n_C}\) provides confusion 
matrix and conventional assessment measures, namely 
the average error rate (ERR), the macro-average precision (PRE), 
the macro-average recall (REC) and the \(F_1\) measure 
(F1 = 2 \cdot \frac{\text{precision \cdot recall}}{\text{precision + recall}}). In addition, the 
average latency estimation (LAET) and the average match 
duration (DUR) are given for the gesture spotting and 
recognition in stream task. When a match is detected, 
the latency and the duration of the match measures are 
defined accordingly to Figure 3. The latency is defined as 
the difference of the matched mid-segment time locations.
5.2. Experimental conditions and Results

For the HMM and HCRF models, following an heuristic approach, the maximal number of iterations relating to the training process was set to 300 and the number of hidden states was initialized to 6, which is adequate for the complexity of the gestures that we consider. The SVM implementation that we have used is a modified version of libsvm (28) able to cope with stream processing. To evaluate comparatively the proposed classification methods, we have performed a subject cross validation experiment consisting in 100 tests: for each test, 10 subjects have been randomly drawn for training and the remaining 10 subjects have been retained for testing. Only the isolated gesture dataset is used for training the classifier. The SVM parameter $\nu$ of the KDTW kernel as well as the SVM meta parameter (RBF bandwidth $\sigma$ and $C$) are optimized using a leave one subject out cross-validation on the train isolated gesture dataset. Nevertheless, the gesture sequence dataset is used for training the classifier. The $\nu$ parameter of the KDTW kernel will be coupled with SVM (29) applied to similar skeleton-based data sets and treating the motion data as undivided whole set. The main limitation of this model, comparatively to the others, is its latency. As already reported in the literature (see for instance (18) for isolated gesture recognition) DTW does not couple well with SVM (the likely cause being its indefiniteness). HCRF performs quite well on the isolated gesture task but is outperformed by the 1NN classifiers for instance (13) for isolated gesture recognition) DTW is its latency. As already reported in the literature (see for instance (18) for isolated gesture recognition) DTW is its latency. As already reported in the literature (see for instance (18) for isolated gesture recognition) DTW is its latency.

### Table 2. Isolated gestures recognition assessment measures

<table>
<thead>
<tr>
<th>Method</th>
<th>ERR mean $\parallel$ std</th>
<th>PRE</th>
<th>REC</th>
<th>F1</th>
<th>LAT</th>
<th>DUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NN DTW</td>
<td>.154 $\parallel$ .012</td>
<td>.869</td>
<td>.866</td>
<td>.867</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1NN KDTW</td>
<td>.128 $\parallel$ .016</td>
<td>.876</td>
<td>.872</td>
<td>.874</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>.318 $\parallel$ .049</td>
<td>.776</td>
<td>.727</td>
<td>.750</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCRF</td>
<td>.128 $\parallel$ .039</td>
<td>.898</td>
<td>.891</td>
<td>.894</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM DTW</td>
<td>.146 $\parallel$ .015</td>
<td>.871</td>
<td>.854</td>
<td>.862</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM KDTW</td>
<td>.051 $\parallel$ .015</td>
<td>.952</td>
<td>.949</td>
<td>.951</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Gestures spotting and recognition in stream assessment measures

<table>
<thead>
<tr>
<th>Method</th>
<th>ERR mean $\parallel$ std</th>
<th>PRE</th>
<th>REC</th>
<th>F1</th>
<th>LAT</th>
<th>DUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NN DTW</td>
<td>.204 $\parallel$ .015</td>
<td>.868</td>
<td>.799</td>
<td>.832</td>
<td>20.9</td>
<td>39.9</td>
</tr>
<tr>
<td>1NN KDTW</td>
<td>.189 $\parallel$ .016</td>
<td>.878</td>
<td>.817</td>
<td>.846</td>
<td>20.9</td>
<td>39.9</td>
</tr>
<tr>
<td>HMM</td>
<td>.404 $\parallel$ .024</td>
<td>.721</td>
<td>.600</td>
<td>.654</td>
<td>4.8</td>
<td>42.4</td>
</tr>
<tr>
<td>HCRF</td>
<td>.244 $\parallel$ .043</td>
<td>.840</td>
<td>.757</td>
<td>.796</td>
<td>16.8</td>
<td>50.8</td>
</tr>
<tr>
<td>SVM DTW</td>
<td>.291 $\parallel$ .086</td>
<td>.783</td>
<td>.714</td>
<td>.741</td>
<td>21.0</td>
<td>50.5</td>
</tr>
<tr>
<td>SVM KDTW</td>
<td>.107 $\parallel$ .021</td>
<td>.947</td>
<td>.893</td>
<td>.920</td>
<td>25.4</td>
<td>41.6</td>
</tr>
</tbody>
</table>

6. Conclusion

We have proposed a framework dedicated to the assessment of gesture spotting and recognition in stream. Despite its dedicated nature, we believe that this framework is quite general purpose and can be used to benchmark continuous pattern detection and recognition in stream. The case study we have carried out using this benchmark version of the DTW measure (KDTW) is particularly efficient on the tested stream data comparatively to the other experimented recognition models (DTW, HMM, HCRF). As a perspective we plan to enhance our excerpt of the NAOPTS dataset in the near future to complete the 24 gesture vocabulary.

Acknowledgments

The authors would like to thank Thales Optronics for their partial support of this research work.

References
