

(c) 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Multidimensional Time Series Shapelets Reliably Detect and Classify Contact Events in Force Measurements of Wiping Actions

Simon Stelter*
stelter@cs.uni-bremen.de

Georg Bartels*,†
georg.bartels@cs.uni-bremen.de

Michael Beetz*
beetz@cs.uni-bremen.de

Abstract—The vision of service robots that autonomously manipulate objects as skillfully and flexibly as humans is still an open challenge. Findings from cognitive psychology suggest that the human brain structures manipulation actions along representations of contact events and their perceptually distinctive sensory signals. In this paper, we investigate how to reliably detect and classify contact events during robotic wiping actions. We present an algorithm that learns the distinct shapes of force measurements during contact events using multidimensional time series shapelets. We evaluate our approach on a dataset consisting of 460 real-world robot wiping episodes that we collected using a table-mounted robot with a wrist-mounted force/torque sensor. Our approach shows good performance with 10-fold cross validation yielding 97.5% precision and 99.3% recall, and can also be used for online contact event detection and classification.

Index Terms—Force and Tactile Sensing, Learning and Adaptive Systems, Service Robots

I. INTRODUCTION

The service robots of the future shall manipulate objects as skillfully and flexibly as humans. Findings from cognitive psychology research suggest that humans achieve high manipulation competence because they mentally structure their actions using representations of events like contact events [1], [2], [3]. Examples of such contact events are the making and breaking of contacts between objects involved in the manipulation. These contact events usually serve as subgoals for action phases that activate and adjust motor controllers, which in turn cause the desired events. More importantly, these contact events bring about discrete and distinct sensory signals. The human brain exploits these distinct sensory signals to reliably monitor the evolution of actions and, in particular, predict the associated contact events [1], [2], [3].

Today’s service robots cannot match the manipulation competence of humans because they lack the necessary perceptual capabilities. One missing skill is the ability to perceive the consequences of their actions, e.g. contact events caused by motions. Without these perception skills, robot programmers are struggling to develop flexible but performant models of robotic manipulation actions. In our research, we aim at transferring insights from cognitive psychology to enable our robots to manipulate objects more skillfully and flexibly, e.g. to perform *robot wiping actions*.

* The authors are all with the Institute for Artificial Intelligence, University of Bremen, Germany.

† Corresponding author.

This work was supported by the EU H2020 project *REFILLS* (Project ID: 731590).

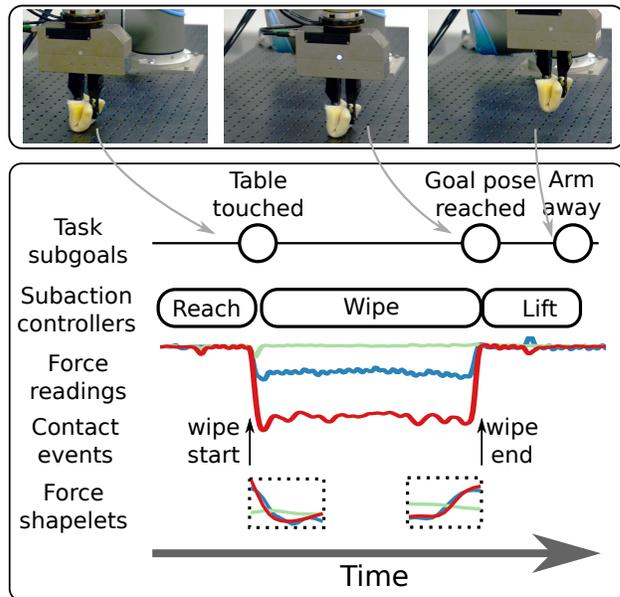


Fig. 1: Representation of a table wiping action structured by contact events, inspired by cognitive psychology [1], [2], [3].

We use the term *robot wiping actions* to refer to actions in which robots move tools along support surfaces to manipulate potential third media in between. These tasks are rich in contact events that trigger either subsequent subactions or error recovery. We consider robotic wiping to be a model problem of robot manipulation.

Figure 1 depicts how we represent an action like wiping a table using contact events. Manipulation actions are structured into subgoals that bound action phases. Action phases activate motor controllers that cause contact events that, in turn, support action monitoring and scheduling because they bring about perceptually distinctive sensory signals.

In this paper, we investigate how to reliably detect and classify contact events during robot wiping actions. We hypothesize that contact events cause perceptually distinctive force measurements and that a learning system can reliably detect contact events using only shape information. Supportingly, other robotics researchers reported that it is easier to detect contact events than contact states from force measurements because the information content of the signals is higher directly after contact events than during contact states [4], [5]. Finally, we assume that our robots use stereotypical motions during wiping to ensure force measurements with distinct shapes.

To investigate our hypothesis, we identified time series shapelets [6] as a promising candidate to capture the distinct shapes of force measurements during contact events and to serve as classifiers. We gathered a dataset comprising 460 real-world robot wiping episodes using a table-mounted robotic manipulator with a wrist-mounted force/torque (F/T) sensor. Within the dataset, we manually identified 10 different contact event classes, and learned a separate shapelet-based classifier for each of them. The learned classifiers show good performance, with 10-fold cross validation yielding 97.5% precision and 99.3% recall. In this paper, we contribute to the state of the art in robotic manipulation in several ways: (1) We demonstrate that contact events during robot wiping tasks cause force measurements with distinct shapes, and that a classifier using only shape information can reliably detect these contact events. (2) We show that multidimensional time series shapelets correctly capture these distinct shapes. (3) We present an algorithm to discover multidimensional time series shapelets that assumes the individual dimensions are dependent. (4) Finally, we present two methods for candidate pruning to greatly speed up the algorithm’s learning phase.

II. RELATED WORK & BACKGROUND

Robotics researchers have already developed various methods to extract contact information from F/T measurements. In early research, hidden Markov models (HMMs) were successfully used to segment the F/T data of teleoperated peg-in-hole tasks into subgoals [7]. However, that approach was evaluated on a small problem set with only four contact states. In another peg-in-hole study, contact events were successfully recognized by applying HMMs on the frequency components of F/T measurements [5]. In a follow-up paper, HMM-based contact detectors were used as process monitors for an industrial assembly task [8].

More recent studies focused on analyzing F/T data in teleoperation or programming by demonstration (PbD) scenarios. During teleoperation, HMMs and Support Vector Machines (SVM) can segment force signals without considering contact information [9]. Force measurements within a PbD framework were successfully segmented using particle filters constrained by contact states [10]. Another PbD study classified contact states of compliant motions using boosting [11]. Continuous force profiles during PbD have been encoded using Gaussian Mixture Models (GMM), but that investigation did not consider any discrete contact information [12]. Neither of these approaches addresses contact event detection and classification. When using HMMs, one cannot visually inspect the time series captured by the model. Time series shapelets, however, offer such visual inspection.

Time series shapelets were introduced as a promising feature for data mining [6]. First learning algorithms focused on 1-dimensional shapelets and combined brute-force search with candidate abandonment to reduce runtime [6]. Since then, other runtime optimizations have been proposed, e.g. computation reuse [13], candidate pruning [13], [14], time series compression [14], [15], and GPU computing [16].

Time series shapelets for multidimensional data have been presented recently [17]. Assuming that individual time series dimensions are independent, ensembles of 1-dimensional shapelet-based decision trees have been shown to outperform ensembles of n -dimensional shapelets [17]. This assumption is not generally valid for F/T data from a single sensor. In fact, our paper presents a shapelet-based algorithm that assumes the dimensions of force data to be dependent.

Recently, deep learning has shown great promise in solving hard robotics problems, e.g tactile material [18] or object classification [19], or tactile slippage detection [20]. A key factor seems to be deep learning’s ability to learn good data representations in an unsupervised fashion [21]. Here, we hypothesize that shapelets are a good representation for the distinct shapes of force signals. If this is true, a generic deep learning algorithm will require more training data than a shapelet-based approach to reach similar performance as it will have to learn a good data representation first.

In our own prior work, we recently published a taxonomy of robotic compliant manipulation tasks that focused on the example of robot wiping actions [22]. We presented a reasoning framework for robot wiping actions that plans wiping motions using a particle-based simulation of the wiping medium [23]. Most recently, we showed how to use haptic perception to infer the effects of wiping motions [24]. With this paper, we contribute to our body of research on robot wiping actions by showing how to use multidimensional time series shapelets to detect contact events in wiping actions.

III. METHODOLOGY

Let us briefly introduce the basic concepts and notations that we use in the remainder of this paper:

A **time series** t_x is a sequence of real values, and $len(t_x)$ is its length. We denote the real value at index i as $t_x[i]$, with the additional requirement $1 \leq i \leq len(t_x)$.

A **time series subsequence** u_x of t_x is denoted by $u_x \sqsubseteq_l t_x$ with $l = len(u_x) \leq len(t_x)$, and is a sequence of consecutive values taken from t_x .

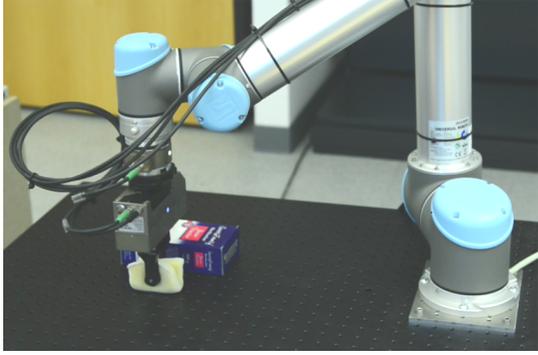
A **multidimensional time series** t is a set of time series, and $dim(t)$ denotes the set of associated dimension labels. If $x \in dim(t)$ is the label of a particular dimension, we call the corresponding time series t_x . We require that $\forall x, y \in dim(t) : len(t_x) = len(t_y) = len(t)$.

A **multidimensional time series subsequence** u of t denoted by $u \sqsubseteq_{(l,d)} t$, is a multidimensional time series with $l = len(u) \leq len(t)$, and $d = dim(u) \subseteq dim(t)$. Additionally, we require that the time series subsequences u_x are generated with a consistent offset index i :

$$\begin{aligned} \forall x \in dim(t), \forall j \in \{1, \dots, len(u_x)\}, \\ \exists i \in \{1, \dots, len(t_x)\} : t_x[i+j] = u_x[j]. \end{aligned}$$

A **multidimensional time series shapelet** (MTS) S is a triple (s, δ, c) , where s denotes a multidimensional time series, c a class, and δ a threshold. If δ is unbound, we write $(s, ?, c)$ ¹.

¹For ease of readability, we will use the terms MTS and shapelet interchangeably in the remainder of this paper.



(a) A table-mounted robot with a wrist-mounted F/T sensor wipes a table with a sponge.



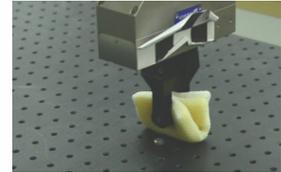
(b) Wiping over an empty table.
Events: $wipe_{\{end\}^{start}}$.



(c) Wiping along a fixated box.
Events: $slide_{\{right\}^{left}}_{\{end\}^{start}}$.



(d) Wiping into a movable box.
Events: $movable_box$.



(e) Wiping over a tightened screw.
Events: $fixed_screw$.

Fig. 2: Experimental setup: Subfigure a) depicts the robot, while subfigures b) - e) show some of the contact events.

Let us briefly outline the rest of this section. First, we describe our experimental setup and learning dataset. Then, we show how to classify, learn, and detect contact events with MTS. Finally, we present candidate pruning techniques.

A. Experimental Setup

Figure 2 depicts our experimental setup. It consisted of a table-mounted 6-DOF manipulator with a F/T sensor mounted to the end of the manipulator and an industrial parallel-jaw gripper. The robot held a soft sponge in its gripper while performing straight-line wiping motion on the table surface. In the depicted experimental setup, we placed a movable object into the path of the robot. We created different environments by placing moveable objects on the table or rigidly attaching objects to the table using screws.

To ensure force signals with distinct shapes our robot performed stereotypical wiping motions. All movements started in a contact-free state above the table. First, the gripper moved down to touch the table. Then, the robot wiped the sponge over the table in a straight line. Across all experiments, we produced contact forces of different maximum magnitudes ($0N - 9.7N$) and trajectories of different lengths ($0.2m - 0.45m$), end-effector speeds ($0.46m/s - 0.55m/s$), and angles between sponge front and wiping direction ($0 - \pi/4$). The experiments ended with the gripper releasing contact and moving back to its starting position.

B. Dataset

Our training dataset D comprises 460 wiping episodes, equaling 93min of meaningful experimental data. Each wiping episode d is a triple (t, C_t, r_t) .

t is a multidimensional time series representing the 3-dimensional force measurement with $dim(t) = \{x, y, z\}$ from the wrist-mounted sensor, downsampled to 25 Hz.² We transformed all t , such that the x-axis points from the beginning of a wiping motion to its end and that the z-axis

²In this study, we focused on obtaining good performance and fast learning times with downsampled force data, only. Future studies investigating more or different contact events should reconsider these choices.

points towards the table surface. We use this automatic data pre-processing to facilitate our learning problem, and can do so because we used only straight-line wiping motions.

C_t is a list of hand-labeled contact events. Figure 2 depicts the 8 contact event classes that we identified: $wipe_{\{end\}^{start}}$, $movable_box$, $fixed_screw$, and $slide_{\{right\}^{left}}_{\{end\}^{start}}$. The classes $slide_{*}$ denote events that may occur when wiping along a fixated box on its left or right side. We also noticed that our algorithm reliably detects long-lived events like $force_{\{dec\}^{inc}}$, even though they could be considered transient contact states. $force_{\{dec\}^{inc}}$ events occur when wiping motions are not parallel to the table plane, and contact forces slowly but constantly increase or decrease, respectively.

For each $c \in C_t$, $r_t(c)$ is a list of hand-labeled time points at which contact events occurred. Please note that we use $r_t(c)$ only during the evaluation, and not during learning.

C. Classification with MTS

An MTS $S = (s, \delta, c)$ can be used as a binary classifier of a multidimensional time series t . To this end, we calculate the best match distance (BMD) between s and t like this:

$$BMD(s, t) = \min_{\substack{u \subseteq (t, a)t \\ l = len(s) \\ d = dim(s)}} \sum_{x \in dim(s)} \sum_{i=1}^{len(s)} \frac{|z(s_x)[i] - z(u_x)[i]|}{len(s) \cdot |dim(s)|} \quad (1)$$

where $z(t_x)$ denotes the normalization of t_x using z-scores:

$$z(t_x)[i] = \begin{cases} \frac{t_x[i] - \mu_{t_x}}{\sigma_{t_x}}, & \text{if } \sigma_{t_x} \geq \sigma_{min} \\ t_x[i] - \mu_{t_x}, & \text{otherwise} \end{cases} \quad (2)$$

σ_{min} is a user-specified parameter, which prevents the amplification of noise. We estimate this parameter by using the maximum standard deviation of subsequences in episodes of the training set where the gripper is not touching anything.

If the BMD is below the threshold δ , t belongs to c . For simplicity, we used the manhattan distance, but it might also be replaced by e.g. the Euclidean distance.

It is important to apply z-normalization to both signals (s_x and u_x) because it ensures that the *BMD* is biased towards similarity in shape. Without it, differences in scale or offset dominate the results.

To allow for the possibility that an event might not be captured in all dimensions of the original measurements, the classifier only uses a subset of all possible dimensions. To detect a collision with a box during wiping, for instance, a shapelet with $dim(s) = \{x, z\}$ proved more reliable than one with $dim(s) = \{x, y, z\}$.

D. Learning MTS

In this subsection we describe how we learn MTS that capture the characteristic shapes of force measurements during contact events. We approach this learning task as a multi-label classification problem using the binary relevance method [25], i.e. we train a separate binary classifier for each contact event class. In total, our learning algorithm has five user-specified parameters: sl_{max} , N_{max} , d_{max} , σ_{min} , and w_{ext} . We describe them when presenting the relevant subalgorithms.

We require our learning algorithm to work with labeled data that we can generate easily. Each label only indicates that a contact event occurred at least once during a wiping episode. For each episode, the learning algorithm does not need to know when or how often a contact event occurred, or how long the candidate shapelets should be.

Algorithm 1: Main loop of the learning algorithm.

```

function find_shapelets(dataset D):
  CLS ← empty dictionary
  C ← {c | (∃(t, Ct, rt) ∈ D) [c ∈ Ct]}
  dims ← {dim | (∃(t, Ct, rt) ∈ D) [dim ∈ dim(t)]}
  SL ← { $\frac{sl_{max} \cdot i}{N_{max}}$  | i ∈ {1..Nmax}}
  foreach c ∈ C, dims' ⊆ dims, sl ∈ SL do
    shapelets ← candidates(D, sl, dims', c)
    foreach S ∈ shapelets do
      Dnew ← features(D, S)
      cls ← train_classifier(Dnew, S)
      CLS[c] ← best_classifier(CLS[c], cls)
  return CLS

```

Algorithm 1 outlines the proposed learning algorithm. For each type of contact event in the dataset, the algorithm searches for the best shapelet to use as a binary classifier. To this end, it exhaustively considers all possible dimensionality subsets and shapelet lengths to create shapelet candidates. This search results in long computation times for a dataset of relevant size. Hence, we devised means of candidate pruning which we describe in subsection III-F. Regarding parameters, sl_{max} specifies the maximum shapelet length, and N_{max} controls how many candidate lengths will be considered.

Algorithm 2 depicts how we calculate a new training dataset D_{new} for each candidate shapelet $S = (s, ?, c)$. To this end, we use the decision whether an episode contains an event of type c to binarize the original dataset D , and employ the *BMD* as a feature extractor to calculate D_{new} .

Algorithm 2: Feature extraction using *BMD*.

```

function features(dataset D, shapelet S = (s, ?, c)):
  Dnew ← empty sequence
  foreach (t, Ct, rt) ∈ D do
    if c ∈ Ct then
      Dnew ← append(Dnew, (BMD(s, t), 1))
    else
      Dnew ← append(Dnew, (BMD(s, t), 0))
  return Dnew

```

Using the new training dataset D_{new} , we complete the new classifier by calculating the distance threshold δ for the candidate shapelet $S = (s, \delta, c)$. To this end, we adapted the method presented by [26] to obtain the algorithm depicted in Algorithm 3.

Algorithm 3: Training of binary classifiers.

```

function train_classifier(training dataset Dnew,
  shapelet S = (s, ?, c)):
  fc ← KDE({d | (∃(d, f) ∈ Dnew) [f = 1]})
  f̄c ← KDE({d | (∃(d, f) ∈ Dnew) [f = 0]})
  pc ← |{d | (∃(d, f) ∈ Dnew) [f = 1]}| / |Dnew|
  p̄c ← |{d | (∃(d, f) ∈ Dnew) [f = 0]}| / |Dnew|
  δ_candidates ← {d ∈ ℝ | Pc(d) = 0.5}
  δ ← argmaxδ' ∈ δ_candidates IG(Dnew, δ')
  return (S = (s, δ, c), i = IG(Dnew, δ), e = fc(δ))

```

We use Gaussian kernel density estimation to calculate $f_c(d)$ and $\bar{f}_c(d)$ that denote the probability densities of a time series t with *BMD* d to be of class c or not, respectively. We then calculate the probability that a time series t is of class c using

$$P_c(d) = \frac{p_c f_c(d)}{\bar{p}_c \bar{f}_c(d) + p_c f_c(d)}, \quad d = BMD(s, t) \quad (3)$$

where p_c and \bar{p}_c denote the corresponding prior probabilities. We use $P_c(d)$ to choose a distance threshold δ at 50% as depicted in Figure 3. If both classes are not as well separated as in this example, there might be multiple points with probability 50%. In that case, we choose the δ with the highest information gain.

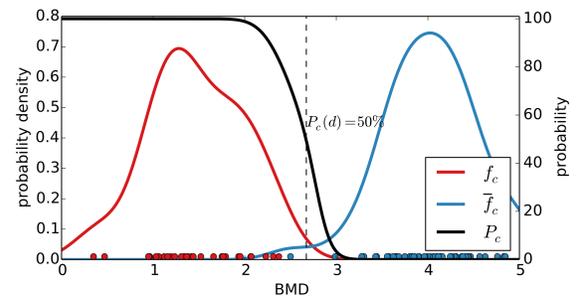


Fig. 3: Selection of threshold δ using the probability $P_c(d)$.

Finally, we choose the better of two candidate classifiers in a similar fashion to selecting the best distance threshold δ at the end of Algorithm 3: The classifier with the higher information gain has priority. If both classifiers happen to have the same information gain, we select the one with the lower probability density at the respective threshold.

E. Contact Event Detection

Algorithm 4 depicts how we use a learned MTS $S = (s, \delta, c)$ to detect contact events of type c in a multidimensional time series t . Using the *BMD* in a moving window approach, we calculate the intermediate time series $d_{t,S}$. Finally, we calculate the relative minima of $d_{t,S}$ with a window length of $len(s)$ to obtain the time indices of contact events. As a result, the algorithm does not detect multiple contact events in close proximity.

Algorithm 4: Detection of contact events using shapelets.

```

function detect(multidimensional time series  $t$ ,
  Shapelet  $S = (s, \delta, c)$ ):
   $d_{t,S} \leftarrow$  empty sequence
  foreach  $u \sqsubseteq_{(len(s), dim(s))} t$  do
     $d_{t,S} \leftarrow$  append( $d_{t,S}, \min(\delta, BMD(s, u))$ )
  return relative_minima( $d_{t,S}, len(s)$ )

```

Figure 4 visualizes this algorithm with an example. The top picture depicts the time series t , the middle plot shows the shapelet S , that we classify with, and the bottom figure visualizes the intermediate distance time series $d_{t,S}$ and the detected time index. There is a time span from approximately 70 to 80, with a *BMD* lower than δ . Because we detect contact events with the relative minima operation, only one time index is selected.

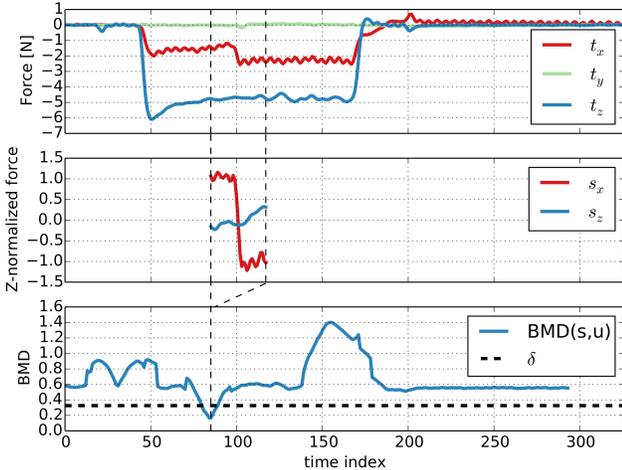


Fig. 4: Visualization of Alg. 4 that detects contact events c in a multidimensional time series t using MTS $S = (s, \delta, c)$.

F. Candidate Pruning

Algorithm 5 depicts how we prune candidate shapelets to speed up the main loop of our learning algorithm. Basically, we employ two strategies for candidate pruning.

Firstly, we hypothesize that contact events bring about force readings with clear extrema either in the raw measurements or its derivatives. Hence, we extract multidimensional time series subsequences that have such extrema in their center. The window length w_{ext} that we use to calculate the relative extrema is a user-specified parameter.

Secondly, we exploit that contact events cause force readings with distinct shapes. To this end, we cluster the candidate shapelets using the *BMD* as a distance measure, and only use the shapelets closest to a cluster center as candidates for learning.

Algorithm 5: Calculation of candidate shapelets.

```

function candidates(dataset  $D$ , shapelet length  $l$ ,
  dimensions  $dim$ , event class  $c$ ):
   $tmp \leftarrow$  empty sequence
   $shapelets \leftarrow \emptyset$ 
  foreach  $t \in \{t \mid (\exists(t, C_t) \in D) [c \in C_t]\}$  do
     $extrema \leftarrow$  find_relative_extrema( $t, dim, w_{ext}$ )
    foreach  $e \in extrema$  do
       $tmp \leftarrow$  append( $tmp, z(subseq(t, l, dim, e))$ )
  foreach  $s \in$  cluster( $tmp$ ) do
     $shapelets \leftarrow$  shapelets  $\cup$  ( $s, ?, l$ )
  return shapelets

```

Algorithm 6 depicts how we prune candidate shapelets using clustering. We first cluster the candidates using the *BMD* as a distance measure. Importantly, we use the mean of all multidimensional time series within a cluster as the center of a cluster. We calculate the mean of a set of multidimensional time series (T) using

$$mean_x(T)[i] = \sum_{t \in T} t_x[i] / |T|. \quad (4)$$

Finally, we choose the 1-nearest neighbors (1-NN) of all cluster centers as candidate shapelets because the mean centers might not look like a real measurement at all. The distance threshold d_{max} for the cluster size is the final user-specified parameter of our learning algorithm.

Algorithm 6: Clustering to prune candidate shapelets.

```

function cluster(sequence of multi. time series  $T$ ,
  threshold  $d_{max}$ ):
  assert( $(\forall t \in T) (\exists l) (\exists d) [len(t) = l \wedge dim(t) = d]$ )
   $centers \leftarrow \emptyset$ 
   $outs \leftarrow T$ 
  while  $outs \neq \emptyset$  do
     $ins \leftarrow \{t \in outs \mid [BMD(t, outs[1]) \leq d_{max}]\}$ 
     $centers \leftarrow centers \cup \{mean(ins)\}$ 
     $outs \leftarrow \{t \in outs \mid (\forall u \in centers) [BMD(t, u) > d_{max}]\}$ 
  return  $\{t \in T \mid (\exists u \in centers) [t = 1-NN(u, T)]\}$ 

```

Because we use the mean of a set of inliers as cluster centers, we avoid using candidates that are close to the border

of a cluster. The authors of [14] presented a clustering-based pruning technique that faced such a problem because it seeded cluster centers with randomly chosen samples.

IV. EVALUATION

We evaluated our method on the dataset described in III-B. For our experiments we used a PC with an Intel 4.0 GHz CPU and 16GB main memory. To ease reproducibility, we released our source code and evaluation dataset at github.com/code-iai/iai_shapelets, branch: RAL17.

A. Contact Event Detection

As every learned MTS constitutes a single-label classifier, we evaluated contact event detection for one label at a time. Given the list of predicted $p_t(c)$ and real events $r_t(c)$ for event class c and time series t , we define the list of correctly predicted events $TP(t, c)$ as described in Algorithm 7.

Algorithm 7: Calculation of true positives TP .

function $TP(\text{multi. time series } t, \text{ event class } c)$:
 $TPs \leftarrow \text{empty dictionary}$
 $sl \leftarrow \text{lookup_shapelet_length}(c)$
foreach $r'_t \in r_t(c)$ **do**
 $p'_t \leftarrow \underset{p_t \in p_t(c)}{\text{argmin}} |p_t - r'_t|$
if $|p'_t - r'_t| < sl$ **and** $|p'_t - r'_t| < |p'_t - TPs[p'_t]|$ **then**
 $TPs[p'_t] \leftarrow r'_t$
return $|TPs|$

Using $TP(t, c)$, we calculate the false positives $FP(t, c)$, false negatives $FN(t, c)$ and true negatives $TN(t, c)$ as

$$FP(t, c) = p_t(c) - TP(t, c) \quad (5)$$

$$FN(t, c) = r_t(c) - TP(t, c) \quad (6)$$

$$TN(t, c) = \text{len}(t) - r_t(c) - FP(t, c). \quad (7)$$

We define true negatives like this because contact events can potentially occur at every point in a time series. Hence, standard accuracy is an unfitting evaluation metric for our domain. Instead, we use precision and recall that yield more meaningful measures. Consider a dummy classifier that never predicts any event at all. It would have an accuracy of almost 100%, but an undefined precision and recall of 0%.

$$\text{precision}(D, c) = \frac{\sum_{t \in D} TP(t, c)}{\sum_{t \in D} (TP(t, c) + FP(t, c))} \quad (8)$$

$$\text{recall}(D, c) = \frac{\sum_{t \in D} TP(t, c)}{\sum_{t \in D} (TP(t, c) + FN(t, c))} \quad (9)$$

Table I depicts average precision, recall, and time differences between the labeled time points and the center of the predicted shapelets for 10-fold cross validation. Average precision and recall are high with 97.5% and 99.3%, respectively. In fact, only the precision of event class *slide_left_end* is low with 85.7%. For the average time difference between

label	number	prec	recall	$\Delta\text{time [s]}$
wipe_start	370	0.966	1.0	0.26
wipe_end	360	0.994	1.0	0.122
force_inc	30	1.0	1.0	0.696
force_dec	30	1.0	1.0	0.452
slide_left_start	30	0.935	0.967	0.132
slide_left_end	30	0.857	1.0	0.103
slide_right_start	30	1.0	1.0	0.0653
slide_right_end	30	1.0	1.0	0.247
movable_box	70	1.0	0.986	0.0603
fixed_screw	160	1.0	0.981	0.0555
average	-	0.975	0.993	0.219

TABLE I: Results from 10 fold cross validation. $d_{max} = .5$, $\sigma_{min} = 0.325$, $w_{ext} = 25$, $N_{max} = 3$, $sl_{max} = 50$.

labeled and predicted contact events we report 0.219s, which is remarkably good because the learning algorithm does not use the time points of labeled contact events. *force_inc* and *force_dec* show the highest time differences, presumably because they resemble transient contact states for which exact time points are harder to determine.

Figure 5 depicts the extracted shapelets when learning from the entire dataset. Regarding *wipe_start*, the algorithm extracted a time series which exhibits an increase in force along $-z$. Intuitively, this makes sense because the z-axis of the measurement frame points towards the wiping surface. In comparison, *wipe_end* looks complementary. However, *wipe_end* contains an additional force decrease along $-x$ that makes the shapelet more specific. *force_inc* is also interesting because it is very long and contains the end of a wipe. Anecdotally, *wipe_end* can be detected within *force_inc*.

B. Candidate Pruning

We evaluated our candidate pruning approaches by comparing the learning times and numbers of shapelet candidates for all possible combinations of pruning techniques. Without pruning, we were forced to train with only 10% of our dataset. Using more data, the algorithm hit the working memory limit during training. We report our results in Table II, with Subtables IIa and IIb depicting our findings when using 10% or 90% of our data for training, respectively. To ensure consistency, we calculated the results of each subtable row using the same training and test datasets.

Subtable IIa shows that both pruning techniques considerably reduce the number of candidates and runtime. Combining both techniques reduces computations even more, while precision and recall are high for all tested scenarios. Please note that we ensured that each contact event class was part of at least 3 episodes of the training set.

Subtable IIb confirms the findings from Subtable IIa: Both pruning techniques reduce the number of candidates, while their combination prunes even more candidates. The number of candidates after clustering are roughly the same for both experiments. Hence, we conclude that the clustering reliably detects the MTS that are associated with the individual contact events. Unfortunately, the learning algorithm did not finish for the non-pruning and extrema-only cases.

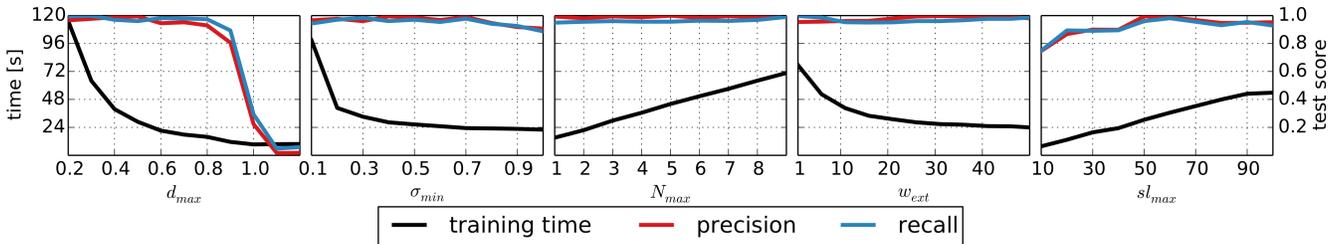


Fig. 6: Influence of the learning parameters on average precision, recall and training time when using 10-fold cross validation. For each subfigure, the fixed parameters were set to $d_{max} = .5$, $\sigma_{min} = .4$, $N_{max} = 3$, $w_{ext} = 25$, $sl_{max} = 50$.

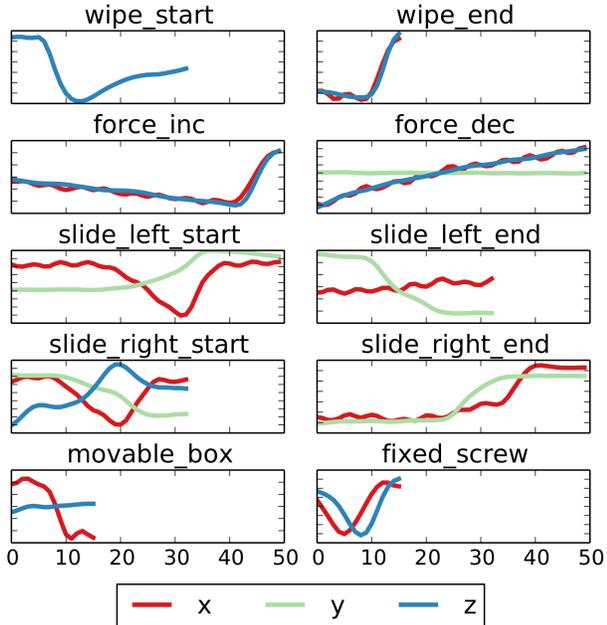


Fig. 5: The shapelets for each event, if the whole dataset is used for training. $d_{max} = .5$, $\sigma_{min} = 0.325$, $N_{max} = 3$, $w_{ext} = 25$, $sl_{max} = 50$.

C. Learning Parameters

Fig. 6 depicts the influence of the learning algorithm’s parameters on average precision, recall and training time when using 10-fold cross validation. For each subfigure, we have fixed 4 of the 5 parameters.

Most parameters are fairly easy to tune. There is clear a trade-off between training time and performance. Only σ_{min} and sl_{max} warrant a brief discussion. Sensor noise is properly filtered out for $\sigma_{min} \in [0.3, 0.7]$, leading to low training times and good performance. Choosing $sl_{max} \in [40, 70]$ ensures short and long shapelet candidates that properly capture all event classes, otherwise performance suffers.

V. DISCUSSION

Let us briefly discuss advantages and limitations of the proposed approach, starting with the advantages.

Our proposed algorithm simultaneously detects and classifies contact events during wiping actions. Furthermore, our

pruning tech.	$ S $ left	runtime [s]	avg prec	avg recall
() extrema	100%	736.418	.932	.967
() cluster	641,361			
(x) extrema	10.9%	90.106	.926	.967
() cluster	69,895			
() extrema	.418%	7.624	.984	.998
(x) cluster	2,678			
(x) extrema	.264%	3.385	.988	.965
(x) cluster	1,695			

(a) 10% train, 90% test split.

pruning tech.	$ S $ left	runtime [s]	avg prec	avg recall
() extrema	100%	n/a	n/a	n/a
() cluster	5,788,041			
(x) extrema	11.1%	approx.	n/a	n/a
() cluster	640,171	4000-5000		
() extrema	.05%	85.926	.957	1.0
(x) cluster	2,893			
(x) extrema	.0342%	27.737	1.0	.967
(x) cluster	1,982			

(b) 90% train, 10% test split.

TABLE II: Evaluation of the pruning techniques. All rows of the same table have used the same training and test set. $d_{max} = .5$, $\sigma_{min} = .4$, $N_{max} = 3$, $w_{ext} = 25$, $sl_{max} = 50$.

learning algorithm is able to discover perceptually distinctive and discriminative shapelets in much longer force measurements even though given labels neither contain any timing information nor number of occurrences. As discussed in [27], most literature on time series classification assumes that the start and end points of patterns of interest can be correctly identified, both during the training and later deployment. Our approach does not make this assumption. Additionally, the discovered shapelets can be visually inspected and intuitively analyzed. This combination of properties makes the algorithm easy to use. Finally, the algorithm can also be used for online event detection.

Of course, the algorithm’s performance relies on having a good training dataset. In particular, we encountered problems with datasets that did not allow a clear separation of two co-occurring event types, e.g. *wipe_start* and *wipe_end* that always appeared together. However, just splitting a single recording in half and adding both to the dataset fixed the issue. In fact, we observed that the performance for one type of event improved when adding new event types that look similar to or contain that event type, e.g. *force_inc*

and *wipe_end*. As a result, we believe that the algorithm scales rather well with increasing numbers of event types and dataset sizes. Unfortunately, long shapelets lead to both longer delays in online detection and increased temporal errors. Hence, future work has to investigate whether the current algorithm is sufficient for reactive control. Potentially, users have to introduce a bias towards shorter shapelets or reduce sl_{max} . Finally, the algorithm's learning times increase exponentially with the number of measurement dimensions. Hence, more optimizations will be required when applying it to time series data with much more than 3 dimensions³.

Several important research questions are outside of the scope of this paper and remain as future work. While we focused on straight-line motions, certain wiping tasks require curved motions, e.g. cleaning the inside of a pot. A possible approach could be to transform the instantaneous force measurements using the known current end-effector velocity to emulate the readings of a straight-line motion. Also, we demonstrated the algorithm's capability to cope with varying end-effector velocities only for a small velocity range. To scale to broader variations, one could use dynamic time warping or shapelet down- or upsampling, respectively. Finally, we did not investigate how shapelets cope with changing friction properties. Our intuition is that a database with sets of shapelets for each contact event could address this problem, while its design is an interesting research topic.

VI. CONCLUSION

In this paper, we presented an algorithm that reliably detects and classifies contact events during robotic wiping actions. Our approach uses MTS to discover and capture the distinct shapes of force measurements caused by contact events. To evaluate our algorithm, we gathered a large dataset of real-world robotic wiping episodes, for which the algorithm yielded good detection and classification performance. Our results showed that MTS can capture the distinct shapes of force measurements, and that MTS can reliably detect and classify contact events during robotic wiping actions. As another contribution, we presented two methods for candidate pruning that greatly improved the speed of learning.

We believe that the presented MTS-based algorithm can be a key component of a haptic perception system for mobile manipulation, and hope that it helps to bring about the service robots of the future that perceive their environments and the consequences of their actions.

REFERENCES

- [1] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Current opinion in neurobiology*, 2006.
- [2] B. Hommel, "Action control according to tec (theory of event coding)," *Psychological Research PRPF*, 2009.
- [3] G. Knoblich and R. Flach, "Predicting the effects of actions: Interactions of perception and action," *Psychological Science*, 2001.
- [4] B. Eberman and J. K. Salisbury, "Application of change detection to dynamic contact sensing," *The International journal of robotics research*, 1994.
- [5] G. Hovland and B. J. McCarragher, "Frequency-domain force measurements for discrete event contact recognition," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996.
- [6] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.
- [7] B. Hannaford and P. Lee, "Hidden markov model analysis of force/torque information in telemanipulation," *The International journal of robotics research*, 1991.
- [8] G. E. Hovland and B. J. McCarragher, "Hidden markov models as a process monitor in robotic assembly," *The International Journal of Robotics Research*, 1998.
- [9] A. Castellani, D. Botturi, M. Bicego, and P. Fiorini, "Hybrid hmm/svm model for the analysis and segmentation of teleoperation tasks," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2004.
- [10] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Transactions on Robotics*, 2007.
- [11] S. Cabras, M. E. Castellanos, and E. Staffetti, "Contact-state classification in human-demonstrated robot compliant motion tasks using the boosting algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2010.
- [12] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, 2011.
- [13] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: an expressive primitive for time series classification," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.
- [14] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme, "Fast classification of univariate and multivariate time series through shapelet discovery," *Knowledge and Information Systems*, 2015.
- [15] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proceedings of the 13th SIAM international conference on data mining*, 2013.
- [16] K.-W. Chang, B. Deka, W. W. Hwu, and D. Roth, "Efficient pattern-based time series classification on gpu," in *IEEE 12th International Conference on Data Mining (ICDM)*, 2012.
- [17] M. S. Cetin, A. Mueen, and V. D. Calhoun, "Shapelet ensemble for multi-dimensional time series," *Quebec: SIAM SDM*, 2015.
- [18] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2016.
- [19] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano, "Tactile object recognition using deep learning and dropout," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.
- [20] M. Meier, F. Patzelt, R. Haschke, and H. J. Ritter, "Tactile convolutional networks for online slip and rotation detection," in *International Conference on Artificial Neural Networks*, 2016.
- [21] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, 2013.
- [22] D. Leidner, C. Borst, A. Dietrich, M. Beetz, and A. Albu-Schäffer, "Classifying compliant manipulation tasks for automated planning in robotics," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015.
- [23] D. Leidner, W. Bejjani, A. Albu-Schäffer, and M. Beetz, "Robotic agents representing, reasoning, and executing wiping tasks for daily household chores," in *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems*, 2016.
- [24] D. Leidner and M. Beetz, "Inferring the effects of wiping motions based on haptic perception," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.
- [25] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.
- [26] Z. Xing, J. Pei, S. Y. Philip, and K. Wang, "Extracting interpretable features for early classification on time series," in *SDM. SIAM*, 2011.
- [27] B. Hu, Y. Chen, and E. Keogh, "Time series classification under more realistic assumptions," in *Proceedings of the SIAM International Conference on Data Mining*, 2013.

³From early experiments we conclude that learning times for 6d F/T measurements are still acceptable on our evaluation computer.