The Version of Record of this manuscript has been published and is available
in the Pattern Recognition Letters journal at https://doi.org/10.1016/j.patrec.2022.11.028

ELSEVIER

Pattern Recognition Letters
journal homepage: www.elsevier.com

TIGER: A Tucker-based Instrument for Gesture Recognition with Inertial Sensors

Laura-Bianca Bilius, Ştefan-Gheorghe Pentiuc, Radu-Daniel Vatavu

MintViz Lab, MANSiD Research Center, Ştefan cel Mare University of Suceava, 13 Universitatii, Suceava 720229, Romania

Article history:

Gesture recognition, Motion gestures, Tensors, Tensor decomposition, Tucker2, Dimensionality reduction, Experiment

ABSTRACT

Gesture input is prevalent for interacting with computer systems, such as mobile and wearable devices. However, representation and acquisition of motion gestures using inertial sensors already built into these devices, *e.g.*, accelerometers and gyroscopes, usually require processing of large training sets because of the high sampling frequency of the collected data. Consequently, one challenge that arises in the classification process of motion gestures for interactive systems is delivering robust and accurate predictions of user input in interactive time. Reducing the size of the training datasets is one way of using more efficiently the computational resources required to run established machine learning algorithms on mobile and wearable devices. To this end, we introduce TIGER, a multilinear tensor-based instrument for motion gesture recognition that (i) uses the Tucker2 decomposition to reduce the dimensionality of the training set by extracting features from the data reported by inertial sensors and (ii) leverages ensemble learning to increase gesture recognition accuracy for devices with built-in inertial sensors.

1. Introduction

A diversity of gesture input has been examined in the scientific literature, including touch, hand, gaze, head, and whole-body gestures [4,12,43,45,19,79,46]. Gesture set design has been informed by documenting users' preferences for gesture input, *e.g.*, in-vehicle interaction [4], smart homes [67], everyday activities [57,37,47], and various devices, *e.g.*, electronic rings [20,69]. Gesture input for such applications can be readily acquired using inertial sensors, *e.g.*, accelerometers and gyroscopes, already embedded in a variety of devices. However, the high sampling rate at which these sensors operate generates large training sets that require proportional storage and processing time for mobile and wearable devices with limited resources. Thus, efficient techniques are needed for *both* representing and recognizing motion gestures on such devices.

We conducted a Systematic Literature Review (SLR) on gesture recognition algorithms trained with data from inertial sensors with a focus on the curse of dimensionality, a phenomenon that leads to low classification accuracy or suboptimal time responses for interactive systems [71]. As prediction and classification algorithms have found a variety of applications in work safety [28], robotics [85], and wearable computing [20], mitigating the course of dimensionality has been reflected by a trade-offbetween response time and classification accuracy.

Based on the findings from our SLR, we propose a new approach for efficient motion gesture recognition using inertial sensors that leverages multilinear tensor representation and decomposition to mitigate the response time vs. recognition accuracy trade-off. Specifically, we employ Tucker decomposition to achieve dimensionality reduction [60] and ensemble learning involving high-performance classifiers informed by our SLR.

Our practical contributions are as follows:

1. A systematic literature review of 52 scientific papers on the topic of gesture recognition using inertial sensors. Our results show that multilinear algebra tensor decomposition was not addressed in any of the papers analyzed in our SLR.
2. TIGER, a new method for motion gesture recognition based on a tensor representation of gestures and ensemble learning with classification approaches informed by our SLR.
3. An evaluation of TIGER using a dataset of 5615 gestures and three different cross-validation techniques.

2. Related Work

2.1. Systematic literature review

To identify relevant references from the scientific literature, we followed Siddaway *et al.*'s [64] recommendations for con-

e-mail: `laura.bilius@usm.ro` (Laura-Bianca Bilius), `pentiuc@usm.ro` (Ştefan-Gheorghe Pentiuc), `radu.vatavu@usm.ro` (Radu-Daniel Vatavu)

Table 1. The processing techniques (total of 167) used by the gesture recognizers from the papers analyzed in our SLR.

Method	Description	References	% (F_q)‡	Mean†	STD†
Sample-rate conversion	Obtain a new discrete unifying representation of input (*e.g.*, decimation, interpolation, resampling)	[63,57,23,73,2,84,81,72,86,37,47,1,56, 24,25,15,51,16,14,50,83,41,82,42]	23.95 (40)	0.870	0.909
Feature scaling	Normalize the data (*e.g.*, normalization, linear min-max scaling)	[57,26,21,10,84,72,86,39,66,15,55,13, 16,50,77,31,58,49,83,41,76,42]	13.77 (23)	0.500	0.548
Filters	Smooth data and remove noise (*e.g.*, high-pass filter, low-pass filter, moving average filter)	[57,10,2,44,84,72,86,47,1,56,75,24,3, 25,15,55,13,16,14,34,77,49,83,82,5,76]	22.01 (37)	0.800	0.900
Feature extraction	Extract relevant information used as features for classification (*e.g.*, Gabor transform, Slope Sign Changes, segmentation, PCA)	[63,26,23,11,21,10,73,2,84,81,72,37, 32,47,39,33,56,52,75,24,3,66,25,55,51, 13,54,34,77,58,49,83,5,76]	40.11 (67)	1.460	1.410
		Total	**167**	**3.212**	**2.023**

†per study (46 papers); ‡F_q=frequency; not specified [30,74,65,59,35,7]

ducting systematic reviews. We ran the following query in the ACM DL, IEEE Xplore, and Scopus electronic databases: *("Document Title": gesture) AND ("Document Title": recognition) AND ("Document Title": IMU OR "Document Title": inertial measurement unit OR "Document Title": Gyroscope OR "Document Title": accelerometer)*. The query returned 71 results, from which we excluded duplicates, keynotes, demonstrations, and entries with abstracts only. The final set of papers used in our analysis consisted of 52 references, from which we extracted information about gesture recognition techniques, dataset size, and gesture dictionaries.

We found that most of the papers have used accelerometers to collect hand motion gestures [63,57,26,10,2,11], but also gyroscopes [21,73,47], EMG [63,57,11], magnetometer [47,58], flexion [59], and contact [59] sensors; see Fig. 1. A few papers have employed multiple sensors, *e.g.*, accelerometers and EMG sensors [83,5,84] or accelerometers and gyroscopes [21,73,41]. The largest dataset contained 15,000 gestures (see Fig. 1) and the size of the gesture dictionaries varied between 1 and 130. We identified 54 methods for preprocessing data acquired from sensors, which we clustered into four categories; see Table 1. The most used method was data normalization (36.5% of the papers analyzed in our SLR) followed by feature extraction (32.7%) and applying a low-pass filter (19.2%). We also identified a number of 47 classification approaches, which we grouped into 11 categories (see Fig. 1): Neural Networks [57, 72,75,3,76], Dynamic Time Warping (DTW) [21,73,2,44], Support Vector Machines (SVMs) [57,23,73], Linear Models (Logistic Regression, Linear Discriminant Analysis) [84, 32,33,41], sequence-based methods (*e.g.*, sign sequence, area sequence) [75,55,77,31], template matching [75,77,65], Hidden Markov Models (HMM) [63,84,83], tree-based classifiers (*e.g.*, Decision Tree, Random Forest) [84,47,58], Nearest Neighbor classification (*e.g.*, k-Nearest Neighbors, Nearest Centroid) [3,66,51,50], Bayesian models (*e.g.*, Naive Bayes, Linear Bayes) [11,3,49], and other (*e.g.*, K-means, Tracking algorithm, Protractor3D, Affinity Propagation, Statistical techniques) [2,84,47,1,75,77,65,41,82,42].

2.2. *Tensorial decomposition*

Tensorial decomposition, as "a way to break the curse of dimensionality" [29], has found a wide range of applications

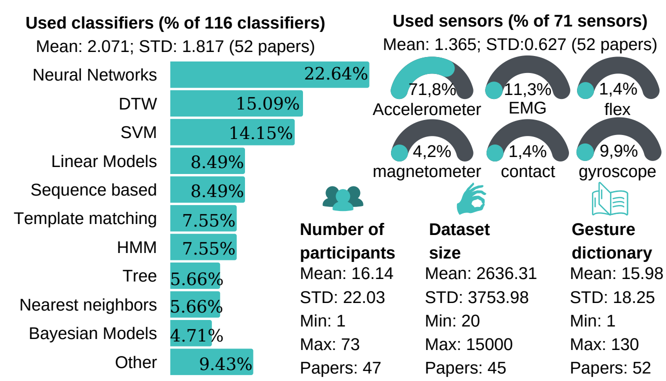

Figure 1. Overview of the classification algorithms, sensors, number of participants, and dataset size from the papers analyzed in our SLR.

in signal processing and machine learning [40]. According to Kolda and Bader [40], tensor analysis and processing consist of two basic methods: decomposing tensors as the sum of a finite number of rank-one tensors (*i.e.*, canonical polyadic decomposition) and decomposing tensors into a core tensor and factor matrices along each mode (*i.e.*, Tucker decomposition). Starting from these basic methods, various decomposition techniques have been proposed, such as PARATUCK2, Block Term Decomposition, or CANDELINC [40]. Among these, the Tucker decomposition results in a core tensor that represents a compressed version of the original [61]. This outcome represents a key feature of this decomposition technique since the core tensor defines "a linking structure among the components of the reduced modes" [17] and retains the intrinsic multidimensional structure of the original data. Moreover, the Tucker decomposition was found to process multidimensional data in a way that captures more of the variance of the data compared to other techniques, such as Principal Component Analysis (PCA) [48].

2.3. *Summary*

Our SLR identified a variety of recognition algorithms for motion gestures employing various data preprocessing techniques. However, since multilinear algebra tensor decomposition was not addressed in any of the papers analyzed in our SLR, we decided to examine and exploit its valuable properties for efficient hand motion gesture recognition; see the next section.

3. TIGER

We introduce TIGER (**T**ucker-based **I**nstrument for **GE**sture **R**ecognition), our new motion gesture recognition method based on the Tucker2 decomposition of data collected from inertial sensors and ensemble learning.

3.1. Gesture tensor representation and decomposition

We represent a motion gesture as the matrix $G_i \in \mathbb{R}^{n \times p_i}$ with $i = \overline{1, N}$, where N is the number of gestures from the training set, n is the number of dimensions (*e.g.*, n=3 for linear accelerations measured along the *x, y,* and *z* axes, n=6 for acceleration and rotation, etc.), and p_i the number of data points in each dimension. Given $G_i \in \mathbb{R}^{n \times p_i}$ and $G_j \in \mathbb{R}^{n \times p_j}$, $p_i, p_j \in \mathbb{N}$, $p_i \neq p_j$, $\forall i \neq j$, i and $j = \overline{1, N}$, a three-way tensor cannot be constructed and, thus, the mean m_g of all gestures G_i is computed, $\forall i = \overline{1, N}$. We then resample each column of gesture G_i m_g/p_i times as the original sample rate [27]. After this resampling step, all the gestures G_i have the same dimensionality $n \times m_g$ and the tensor $T_G \in \mathbb{R}^{n \times m_g \times N}$ can be constructed successfully; see Algorithm 1 and [40] for details.

Tucker's decomposition [40] is then applied to T_G to reduce its dimensionality and extract relevant features. The result is represented by a core tensor and multiple factor matrices that correspond to each dimension of the core tensor, as follows[1]:

$$T_G \approx T_g \times_1 A \times_2 B \times_3 C = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_p \circ b_q \circ c_r = [\![T_g; A, B, C]\!] \tag{1}$$

where $T_g \in \mathbb{R}^{P \times Q \times R}$ is the core tensor that can be seen as a compressed version of the original T_G with values that reveal the interaction between its components, $A \in \mathbb{R}^{n \times P}$, $B \in \mathbb{R}^{m_g \times Q}$, and $C \in \mathbb{R}^{N \times R}$ are factor matrices corresponding to each dimension, and P=n, Q=1, and R=N are the number of components in the factor matrices. The factor matrices can also be seen as principal components along the different dimensions of the data [40]. Our method does not decompose along the dimension that corresponds to the number of gestures to reduce only those dimensions corresponding to n and m_g. For a three-way tensor $T_G \in \mathbb{R}^{n \times m_g \times N}$, the Tucker2 decomposition is defined as follows:

$$T_G = T_g \times_1 A \times_2 B = [\![T_g; A, B, I]\!] \text{ where } C = \mathbf{I} \tag{2}$$

We choose the number of Tucker2 components [40] for m_g and n to be 1 and n in order to reduce each gesture data to an array with a size equal to the number of different sensor measurements. As shown in Fig. 2, the gesture tensor T_G is decomposed into a core tensor $T_g \in \mathbb{R}^{n \times 1 \times N}$ and factor matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{1 \times m_g}$. The corresponding factor matrix for the N dimension is the identity matrix I and the m_g and n dimensions are decomposed into n and 1 components, respectively. The core of the tensor $T_g \in \mathbb{R}^{n \times 1 \times N}$ is a matrix denoted by $M_g \in \mathbb{R}^{n \times N}$ where each gesture is a vector of length n.

[1]The symbol "$\times_n$" is the n-mode product, *i.e.* represents the multiplication of a tensor with a matrix or a vector in n-mode [40] and the symbol "$\circ$" represents the outer product of vectors [40].

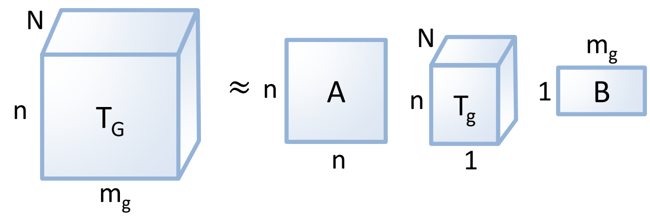

Figure 2. Tucker2 decomposition of the gesture tensor $T_G \in \mathbb{R}^{n \times m_g \times N}$.

Algorithm 1 Construction of the gesture tensor.

Input: $G_i \in \mathbb{R}^{n \times p_i}, i = \overline{1, N}$ *gestures*
Output: T_g, A, B
for $i \leftarrow 1$ *to* N **do**
  for $j \leftarrow 1$ *to* n **do**
    $G_i[:, j]$ = moving_average_filter(resample($G_i[:, j]$))
  end
end
$T_G = [G_1, G_2, G_3, \cdots, G_N]$, *where size of* $T_G = [n, m_g, N]$
// apply Tucker2 on gesture tensor T_G
$[T_g, A, B] = T_G \times_1 A \times_2 B, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{1 \times m_g}, T_g \in \mathbb{R}^{n \times 1 \times N}$

3.2. Optimization of prediction accuracy

Ensemble learning represents a technique to increase classification accuracy by involving several independent classifiers and voting [22]. The classifiers can be chosen by the practitioner or following an optimal cross-validation accuracy evaluation with hyper-parameter optimization [80]. In our case, we inform the structure of the ensemble using both approaches: we select the best performing motion gesture classifiers from the analysis of our SLR and run cross-validation performance evaluation to confirm our choice; see the next section for details.

Once the ensemble is defined, the next step consists in the preprocessing of the gesture candidate that is subject to classification. The gesture is resampled to obtain a matrix of dimension $G_{N+1} \in \mathbb{R}^{n \times m_g}$ that is multiplied with the factor matrices[2]:

$$g_{N+1} = (G_{N+1} \cdot A)^T \cdot B, g_{N+1} \in \mathbb{R}^n, G_{N+1} \in \mathbb{R}^{n \times g_{N+1}} \tag{3}$$

to obtain a new representation that is linearly transformed in a space of lower dimensionality. The gesture is then classified by a voting committee composed of c independent classifiers, and the class occurring with the highest frequency, f wins the vote. For cases where tie-breaking is needed, we employ the k-fold cross-validation misclassification mean error of the highest accuracy values on the training data as the tie-break rule [62]:

$$E = [E_1, E_2, ..., E_q], E_i = \left[\frac{\sum_{t=1}^{c} e(t | l_{fi} = l_t)}{f} \right] \tag{4}$$

where $l_f = [l_{f1}, l_{f2}, \cdots, l_{fq}]$ represents the array of the q class occurring with the highest frequency f, e=$[e_1, e_2, ..., e_c]$ is an array of the k-fold cross-validation misclassification errors of

[2]The symbol "$\cdot$" represents the matrix multiplication.

Algorithm 2 Gesture recognition algorithm using tensor decomposition and ensemble voting.

Input: *T_g, A, B, and test gesture G_{N+1}*
Output: *label L of gesture G_{N+1}*
for $i \leftarrow 1$ *to* n **do**
  $G_{N+1}[:, i]$ = moving_average_filter(resample($G_{N+1}[:, i]$))
end
$g_{N+1} = (G_{N+1} \cdot A)^T \cdot B$
Model = fit($[C_1, C_2, \cdots, C_c]$, T_g)
e = missclassification_error(fit($[C_1, C_2, \cdots, C_c]$))
$e : [e_1, e_2, \cdots, e_c]$
$[l_1, l_2, \cdots, l_c]$ = predict (Model, g_{N+1})
$[f]$= max (frequency($[l_1, l_2, \cdots, l_c]$))
if *length* $[f] > 1$ **then**
  foreach $k \in [l_{f1}, l_{f2}, \cdots, l_{fq}]$ **do**
    $s = 0$ *, where s represents the sum of missclassification errors of the highest frequency labels*
    foreach $t \in [l_1, l_2, \cdots, l_c]$ **do**
      if $l_{fk} == l_t$ **then**
        $s = s + e_t$
      end
    end
    $E[k] = s/f$
  end
  $L = l_{min(E)}$
else
  $L = l_f$,
  where l_f is the class occurring with the highest frequency
end

classifiers $C_1, ..., C_c$, E represents the array of the q mean errors of classes occurring with the highest frequency f, and $l = [l_1, l_2, \cdots, l_c]$ are the classes assigned by classifiers $C_1, ..., C_c$ to a gesture. The label that corresponds to the smallest error is represented by $min(E)$; see Algorithm 2.

4. Results

To evaluate the recognition accuracy of our method, we used the 6DMG dataset consisting of 5615 gestures of 20 distinct types collected from 28 participants using the Wiimote built-in accelerometer and gyroscope sensors [8,9]; see Fig. 3. We preferred this dataset for our evaluation because it includes a wide range of motion gestures commonly employed for interactive systems, such as directional swipes and geometrical shapes. Also, the number of distinct gestures from this dataset is larger than the average gesture set size from the scientific literature of gesture input with electronic rings [69].

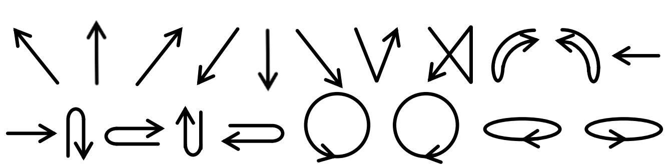

Figure 3. Gesture types of the 6DMG dataset [8,9] used in our evaluation.

Table 2. The top-20 algorithmic variants evaluated with grid search hyper-parameter optimization and 5-fold cross-validation; see also Fig. 1 for our SLR results. The top-5 algorithms were included in our ensemble.

	Classifier	$\mu^{\ddagger}$	**Std**$^{\curlywedge}$
C_1	**SVM† (ovr, rbf kernel)**	90.90	6.43
C_2	**Extra Trees (50 estimators, 2 split)**	90.85	5.82
C_3	**SVM (ovr, linear kernel)**	90.01	5.95
C_4	**SVM† (ovr, linear kernel)**	89.12	6.93
C_5	**Extra Trees (20 estimators, 2 split)**	88.98	6.08
C_6	Label Propagation (rbf kernel)	87.57	8.77
C_7	Label Spreading (rbf kernel)	87.53	8.75
C_8	1NN† (Euclidean distance)	86.87	9.17
C_9	1NN† (correlation distance)	86.24	9.01
C_{10}	1NN (correlation distance)	85.76	6.67
C_{11}	SVM (ovr, rbf kernel)	85.58	5.64
C_{12}	Bagging Classifier (10 SVM estimators)	85.43	5.28
C_{13}	5NN† (Euclidean distance)	84.72	8.17
C_{14}	Logistic Regression† (normalized data)	84.12	8.34
C_{15}	3NN (Canberra distance)	83.87	6.36
C_{16}	5NN (correlation distance)	83.74	6.49
C_{17}	1NN (Euclidean distance)	83.73	6.57
C_{18}	15NN (Manhattan distance)	82.27	6.37
C_{19}	5NN (Euclidean distance)	81.91	6.34
C_{20}	15NN (correlation distance)	81.29	6.39

†Standardized data; ‡Mean; $^{\curlywedge}$Standard deviation.

To construct the tensor T_G, we resampled the gestures from the 6DMG dataset and applied a moving average filter to smooth the sensor signals. We represented each gesture with a 13×69 matrix and used a tensor of size $13 \times 69 \times 5,615$ for the entire dataset. The Tucker2 (Eq. 2) with constrained singular value decomposition [6] returned a $5,615 \times 13$ matrix. To compile the voting committee of the ensemble, we conducted grid search hyper-parameter optimization [80] with cross-validation and evaluated 20 algorithmic variants from scikit-learn [18]; see Table 2 for the top twenty best performing classifiers. From these, we selected the five best performing gesture classifiers, denoted with C_1, C_2, C_3, C_4 and C_5 in the following, to constitute our ensemble.

We employed leave-one-gesture-out, a subject dependent scenario (Table 3), leave-one-subject-out, a subject independent scenario (Table 4), and a leave-one-subject-out cross-validation technique function of the number of participants from which gesture samples were collected for training (Fig. 4) to evaluate the recognition accuracy of our method. The latter cross-validation technique was employed to evaluate the effect of the dataset dimensionality on recognition accuracy. For comparison purposes, we include in our evaluation two other techniques that have been employed in the scientific literature for dimensionality reduction (see Table 1 from our SLR): Principal Component Analysis (PCA) [53] and Nonnegative Matrix Factorization (NNMF) [38], which represent two variants of Singular Value Decomposition (SVD). PCA and NNMF were applied on the resampled, filtered, and matricized dataset that was decomposed into n components to make a fair comparison with

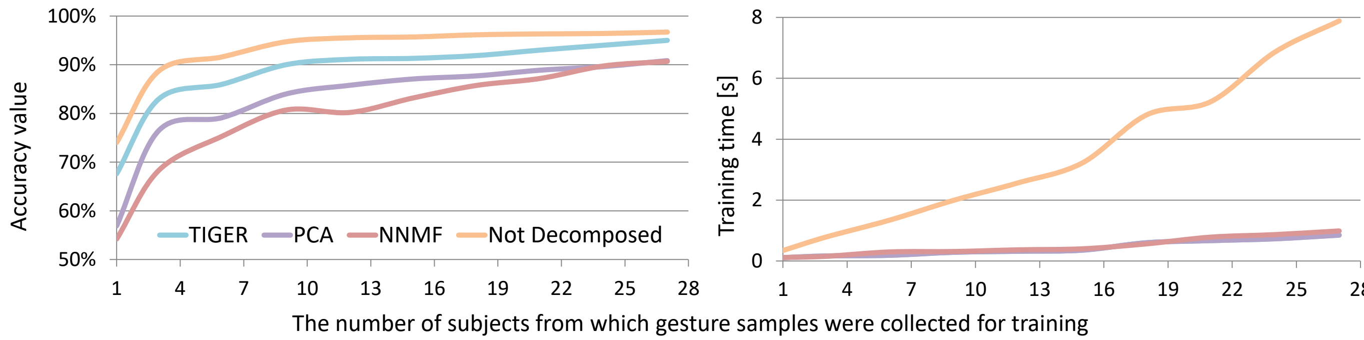

Figure 4. Recognition accuracy rates obtained with leave-one-subject-out cross-validation technique function of the number of participants (left) and the training time (right) function of the number of participants from which gesture samples were collected for training.

Table 3. Recognition accuracy rates obtained using the leave-one-gesture-out cross-validation (LOGO-CV) procedure.

Model	TIGER	PCA	NNMF	ND†
Decomposition time (s)	0.877	0.321	124.34	–
Learning time (s)	0.753	0.825	0.105	8.127
Classification time (s)	1.351	1.456	1.313	2.003
Recognition accuracy C_1	97.5%	97.1%	90.1%	99.3%
Recognition accuracy C_2	95.6%	96.4%	86.1%	99.2%
Recognition accuracy C_3	95.8%	96.3%	91.1%	98.5%
Recognition accuracy C_4	96.7%	97.2%	89.7%	99.1%
Recognition accuracy C_5	97.8%	96.8%	86.4%	98.3%
Ensemble voting	**99.1%**	**97.7%**	**91.5%**	**99.6%**

The Tucker2 sum of squared error was 35.21.
The sum of variances of the first 13 PCA components was 89.2%.
The NNMF root mean square residual was 0.91.
†ND = non-decomposed data.

Table 4. Recognition accuracy rates obtained with the leave-one-subject-out cross-validation (LOSO-CV) procedure.

Model	TIGER	PCA	NNMF	ND†
Decomposition time (s)	1.144	0.963	105.46	—
Learning time (s)	0.701	0.722	0.988	7.887
Classification time (s)	1.234	1.457	1.282	1.933
Recognition accuracy C_1	85.1%	86.2%	72.8%	96.1%
Recognition accuracy C_2	88.5%	87.1%	64.6%	95.8%
Recognition accuracy C_3	88.9%	86.5%	59.4%	95.2%
Recognition accuracy C_4	87.8%	86.4%	75.7%	94.9%
Recognition accuracy C_5	89.5%	86.9%	57.9%	95.6%
Ensemble voting	**92.9%**	**89.9%**	**76.6%**	**96.2%**

The Tucker2 sum of squared error was 38.39.
The sum of variances of the first 13 PCA components was 88.5%.
The NNMF root mean square residual was 0.89.
†ND = non-decomposed data.

TIGER, but we also evaluated recognition accuracy for non-decomposed data. Moreover, TIGER consists of both Tucker2 decomposition and ensemble learning, but the ensemble was treated separately in this study for PCA, NNMF, and the non-decomposed data to appreciate the effect of the dimensionality reduction achieved with the Tucker2 decomposition.

Tables 3 and 4 list the recognition accuracy rates obtained for ensemble learning and each individual classifier C_1, C_2, C_3, C_4, and C_5 with leave-one-gesture-out and leave-one-subject-out cross-validation with ensemble learning delivering the higher accuracy. Also, the decomposition time was smaller for Tucker and PCA, but significantly larger for NNMF. After the Tucker2 decomposition, the training data represented only 1.45% of the original dataset with more training time needed for the non-decomposed data, but lower for TIGER; see Table 4.

We studied the effect of training data size on gestures recognition accuracy by employing the leave-one-subject-out cross-validation technique. We performed S=28 training rounds and for each round, we trained the ensemble using $S - O$ subjects data, where $O = \overline{1, S-1}$, and S represent the total number of participants. Similar results as LOSO-CV (see Table 4) were obtained using the latter technique, leave-one-subject-out cross-validation function of the number of participants from which gesture samples were collected for training; see Fig. 4, left. The trends from Fig. 4 indicate that recognition accuracy rates increase when gesture samples are collected from more participants. Also, training time increases significantly for non-decomposed data compared to the dimensionality techniques discussed in this paper; see Fig. 4, right.

5. Limitations

We found that using the Tucker2 decomposition for feature extraction and dimensionality reduction led to smaller training and classification time without compromising recognition accuracy. This result recommends our method for motion gesture recognition within the constraints of interactive response time. Also, TIGER offers valuable advantages of reducing computational cost when training datasets are large, but, for training datasets of smaller size, the recognition accuracy was slightly lower compared to the non-decomposed data. This finding led to our design decision to employ ensemble learning for TIGER to achieve high recognition accuracy.

Gesture input with mobile and wearable devices can be encumbered by everyday situations, such as driving or walking. TIGER is intended for user-segmented gesture input, *e.g.*, via button presses [36], and was not evaluated for continuous gesture recognition, which we leave for future work. Nevertheless, user-segmented input has the net advantage of empowering users with the feeling of control over input, eliminates the

Midas effect affecting gesture-based interaction, and is more robust compared to automatic gesture segmentation approaches.

6. Conclusion

We conducted a systematic literature review to analyze current recognition approaches for motion gestures that use inertial sensors. Based on our findings, we proposed TIGER, a new recognition method that leverages multilinear algebra, tensors for gesture representation, Tucker2 decomposition for both feature extraction and dimensionality reduction of gesture data. Coupled with ensemble learning, our method surpassed existing techniques in both recognition accuracy and training and classification time. To the best of our knowledge, our work is the first applying tensor decomposition for hand motion gesture recognition acquired using inertial sensors. We see more opportunities of employing multilinear tensor algebra decomposition for gesture recognition, including automatic gesture segmentation in continuous gesture data, combining statistical classifiers that use gesture features with Tucker2 decomposition, but also extending our method for other gesture types, such as gestures of the head [70], feet [78], and whole body [68].

Acknowledgements

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, PN-III-P3-3.6-H2020-2020-0034 (12/2021).

References

[1] A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, compressive sensing. In *Int. Conf. on Acoustics, Speech and Signal Processing*, 2010.

[2] A. Akl, C. Feng, and S. Valaee. A Novel Accelerometer-Based Gesture Recognition System. In *IEEE Transactions on Signal Processing*, 2011.

[3] D. O. Anderez, L. P. Dos Santos, A. Lotfi, and S. W. Yahaya. Accelerometer-based Hand Gesture Recognition for Human-Robot Interaction. In *SSCI*, 2019.

[4] L.-B. Bilius and R.-D. Vatavu. A multistudy investigation of drivers and passengers' gesture and voice input preferences for in-vehicle interactions. In *J. of Intelligent Transportation Systems, Taylor & Francis*, 2021.

[5] M. B. Brahem, B.-A. J. Ménélas, and M. J.-D. Otis. Use of a 3DOF Accelerometer for Foot Tracking and Gesture Recognition in Mobile HCI. *Procedia Computer Science*, 19, 2013.

[6] S. Brunton and J. Kutz. Singular Value Decomposition (SVD). 2019.

[7] M. Chakravarthi, R. K. Tiwari, and S. Handa. Accelerometer Based Static Gesture Recognition and Mobile Monitoring System Using Neural Networks. *Procedia Computer Science*, 70, 2015.

[8] M. Chen, G. AlRegib, and B.-H. Juang. A new 6D motion gesture database and the benchmark results of feature-based statistical recognition. In *IEEE Int. Conf. Emerging Signal Processing Applications*, 2012.

[9] M. Chen, G. AlRegib, and B.-H. Juang. 6DMG: A New 6D Motion Gesture Database. In *MMSys*. ACM, 2012.

[10] W. Chen and R. Lyu. A HMM-based fundamental motion synthesis approach for gesture recognition on a nintendo triaxial accelerometer. In *ICSPCS*, 2011.

[11] X. Chen, X. Zhang, Z. Zhao, J. Yang, V. Lantz, and K. Wang. Hand Gesture Recognition Research Based on Surface EMG Sensors and 2D-accelerometers. In *Int. Symp. on Wearable Computers*, 2007.

[12] Y. Cheng, G. Li, M. Yu, D. Jiang, J. Yun, Y. Liu, Y. Liu, and D. Chen. Gesture recognition based on surface electromyography-feature image. *Concurrency and Computation: Practice and Experience*, 33, 10 2020.

[13] H. S. Chudgar, S. Mukherjee, and K. Sharma. S control: Accelerometer-based gesture recognition for media control. In *Int. Conf. on Advances in Electronics Computers and Communications*, 2014.

[14] K. Dermitzakis, A. H. Arieta, and R. Pfeifer. Gesture recognition in upper-limb prosthetics: A viability study using dynamic time warping and gyroscopes. In *IEEE EMBS*, 2011.

[15] J. Ducloux, P. Colla, P. Petrashin, W. Lancioni, and L. Toledo. Accelerometer-based hand gesture recognition system for interaction in digital TV. In *Proc. of I2MTC*, 2014.

[16] J. Ducloux, P. Petrashin, W. Lancioni, and L. Toledo. Remote control with accelerometer-based hand gesture recognition for interaction in digital TV. In *EAMTA*, 2014.

[17] E. Ceulemans, H. Kiers. Selecting among three-mode principal component models of different types and complexities: a numerical convex hull based method. *The British J. of math. and stat. psychology*, 2006.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. Scikit-learn: Machine Learning in Python. *in Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Q. Gao, J. Liu, Z. Ju, and X. Zhang. Dual-Hand Detection for Human–Robot Interaction by a Parallel Network Based on Hand Detection and Body Pose Estimation. *IEEE Transactions on Industrial Electronics*, 66(12):9663–9672, 2019.

[20] B. Gheran, J. Vanderdonckt, and R. D. Vatavu. Gestures for Smart Rings: Empirical Results, Insights, and Design Implications. In *DIS*, 2018.

[21] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma. A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. In *IEEE Sensors Journal*, volume 16, pages 6425–6432, 2016.

[22] I. Harris K Butler, M. A. Friend, J. Kenneth W Bauer, and T. J. Bihl. The effectiveness of using diversity to select multiple classifier systems with varying classification thresholds. In *JACT*, 2018.

[23] Z. He. A New Feature Fusion Method for Gesture Recognition Based on 3D Accelerometer. In *Conf. on Pattern Recognition (CCPR)*, 2010.

[24] Z. He. Gesture Recognition Based on Tri-Axis Accelerometer Using 1D Gabor Filters. In *PAAP*, 2018.

[25] Z. He and Z. He. Boosting Gabor Feature Extraction for Gesture Recognition using Tri-axis Accelerometer. In *Information Technology and Mechatronics Engineering Conference*, 2018.

[26] Z. He and Z. He. Accelerometer-based Gesture Recognition Using MFCC and HMM. In *ICCC*, 2018.

[27] MATLAB & Simulink. MathWorks. 2021.

[28] L. Huang, Q. Fu, M. He, D. Jiang, and Z. Hao. Detection algorithm of safety helmet wearing based on deep learning. *Concurrency and Computation: Practice and Experience*, 33, 2021.

[29] I. Kisil, G. Calvi, B. Scalzo Dees, D. Mandic. *Tensor Decompositions and Practical Applications: A Hands-on Tutorial*. Springer International Publishing, Cham, 2020.

[30] R. Izuta, K. Murao, T. Terada, and M. Tsukamoto. Early Gesture Recognition Method with an Accelerometer. In *MoMM*. ACM, 2014.

[31] D. Jayaraman and K. Vanitha. Nonspecific-user hand gesture recognition by using MEMS accelerometer. In *Int. Conf. on Info. Comm. and Embedded Syst. (ICICES)*, 2014.

[32] S. Jiang, B. Lv, X. Sheng, C. Zhang, H. Wang, and P. B. Shull. Development of a real-time hand gesture recognition wristband based on sEMG and IMU sensing. In *ROBIO*, 2016.

[33] S. Jiang, B. Lv, W. Guo, C. Zhang, H. Wang, X. Sheng, and P. B. Shull. Feasibility of Wrist-Worn, Real-Time Hand, and Surface Gesture Recognition via sEMG and IMU Sensing. In *IEEE Transactions on Industrial Informatics*, volume 14, pages 3376–3385, 2018.

[34] M. Joselli and E. Clua. gRmobile: A Framework for Touch and Accelerometer Gesture Recognition for Mobile Games. In *Symp. on Games and Digital Entertainment*, 2009.

[35] A. Kannan, A. Ramesh, L. Srinivasan, and V. Vijayaraghavan. Low-cost static gesture recognition system using MEMS accelerometers. In *Global Internet of Things Summit (GIoTS)*, 2017.

[36] H. Ketabdar, M. Roshandel, and K. A. Yüksel. Towards Using Embedded Magnetic Field Sensor for around Mobile Device 3D Interaction. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, page 153–156. Association for Computing Machinery, 2010.

[37] M. Khan, S. I. Ahamed, M. Rahman, and J. Yang. Gesthaar: An accelerometer-based gesture recognition method and its application in NUI driven pervasive healthcare. In *ICESPA*, 2012.

[38] J. Kim and H. Park. Sparse Nonnegative Matrix Factorization for Clustering. 2008.

[39] P. Koch, M. Dreier, M. Maass, M. Böhme, H. Phan, and A. Mertins. A Recurrent Neural Network for Hand Gesture Recognition based on Accelerometer Data. In *Int. Conf. EMBC*, 2019.

[40] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. In *SIAM Review*, 2009.

[41] S. Kratz, M. Rohs, and G. Essl. Combining Acceleration and Gyroscope Data for Motion Gesture Recognition Using Classifiers with Dimensionality Constraints. In *IUI '13*. ACM, 2013.

[42] B. M. Lee-Cosio, C. Delgado-Mata, and J. Ibanez. ANN for Gesture Recognition using Accelerometer Data. *Procedia Technology*, 3, 2012.

[43] S. Liao, G. Li, H. Wu, D. Jiang, Y. Liu, J. Yun, Y. Liu, and D. Zhou. Occlusion gesture recognition based on improved SSD. *Concurrency and Computation: Practice and Experience*, 33, 10 2020.

[44] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. In *Int. Conf. on Pervasive Computing and Communications*, 2009.

[45] Y. Liu, D. Jiang, H. Duan, Y. Sun, G. Li, B. Tao, J. Yun, Y. Liu, and B. Chen. Dynamic Gesture Recognition Algorithm Based on 3D Convolutional Neural Network. *Comp. Intell. and Neuroscience*, 2021.

[46] Y. Liu, C. Li, D. Jiang, B. Chen, N. Sun, Y. Cao, B. Tao, and G. Li. Wrist angle prediction under different loads based on GA-ELM neural network and surface electromyography. *Concurrency and Computation: Practice and Experience*, 34(3), 2021.

[47] M. Kang, H. Kang, C. Lee, K. Moon. The gesture recognition technology based on IMU sensor for personal active spinning. In *ICACT*, 2018.

[48] M. Mansi. Comparing Multilinear PCA and Kernel PCA with Linear PCA technique. *Int. Journal of Distributed Sensor Networks*, 16(4), 2020.

[49] D. Mace, W. Gao, and A. Coskun. Accelerometer-Based Hand Gesture Recognition Using Feature Weighted Naive Bayesian Classifiers and Dynamic Time Warping. In *Proc. UIU'13 Companion*. ACM, 2013.

[50] O. Makaussov, M. Krassavin, M. Zhabinets, and S. Fazli. A Low-Cost, IMU-Based Real-Time On Device Gesture Recognition Glove. In *Int. Conf. SMC*, 2020.

[51] T. Marasović and V. Papić. Accelerometer based gesture recognition system using distance metric learning for nearest neighbour classification. In *Int. Workshop on Machine Learning for Signal Processing*, 2012.

[52] G. Marqués and K. Basterretxea. Efficient Algorithms for Accelerometer-Based Wearable Hand Gesture Recognition Systems. In *Int. Cont. Embedded and Ubiquitous Computing*, 2015.

[53] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram. Multivariate Statistical Data Analysis-Principal Component Analysis. In *Int. J. of Livestock Research*, 2017.

[54] F. Moiz, P. Natoo, R. Derakhshani, and W. D. Leon-Salas. A comparative study of classification methods for gesture recognition using a 3-axis accelerometer. In *International Conference on Neural Networks*, 2011.

[55] L. Nguyen-Dinh, D. Roggen, A. Calatroni, and G. Tröster. Improving online gesture recognition with template matching methods in accelerometer data. In *Int. Conf. ISDA*, 2012.

[56] G. Niezen and G. P. Hancke. Evaluating and optimising accelerometer-based gesture recognition techniques for mobile devices. In *AFRICON 2009*, pages 1–6, 2009.

[57] T. Pan, W. Tsai, C. Chang, C. Yeh, and M. Hu. A Hierarchical Hand Gesture Recognition Framework for Sports Referee Training-Based EMG and Accelerometer Sensors. In *IEEE Transactions on Cybernetics*, 2020.

[58] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis. RisQ: Recognizing Smoking Gestures with Inertial Sensors on a Wristband. In *MobiSys '14*. ACM, 2014.

[59] V. Pathak, S. Mongia, and G. Chitranshi. A framework for hand gesture recognition based on fusion of Flex, Contact and accelerometer sensor. In *ICIIP*, 2015.

[60] A.-H. Phan and A. Cichocki. Tensor decompositions for feature extraction and classification of high dimensional datasets. In *Nonlinear Theory and Its Applications, IEICE*, volume 1, 10 2010.

[61] S. Zafeiriou. Algorithms for Nonnegative Tensor Factorization. In *Tensors in Image Processing and Computer Vision*, 2009.

[62] C. Sammut and G. I. Webb, editors. *Leave-One-Out Cross-Validation*, pages 600–601. Springer, 2010.

[63] S. Shin, D. Kim, and Y. Seo. Controlling Mobile Robot Using IMU and EMG Sensor-Based Gesture Recognition. In *Int. Conf. on Broadband and Wireless Computing, Communication and Applications*, 2014.

[64] A. P. Siddaway, A. M. Wood, and L. V. Hedges. How to Do a Systematic Review: A Best Practice Guide for Conducting and Reporting Narrative Reviews, Meta-Analyses, and Meta-Syntheses. In *Annual Review of Psychology*, volume 70(1), pages 747–770, 2019.

[65] O. Sidek and M. Abdul Hadi. Wireless gesture recognition system using MEMS accelerometer. In *Int. Symp. on Technology Management and Emerging Technologies*, 2014.

[66] T. Marasović, V. Papić. Feature weighted nearest neighbour classification for accelerometer-based gesture recognition. In *IC SoftCOM*, 2012.

[67] R.-D. Vatavu. A Comparative Study of User-defined Handheld Vs. Freehand Gestures for Home Entertainment Environments. In *Journal of Ambient Intelligence and Smart Environments*, volume 5(2), 2013.

[68] R.-D. Vatavu. Beyond features for recognition: Human-readable measures to understand users' whole-body gesture performance. *International Journal of Human–Computer Interaction*, 33(9):713–730, 2017. doi: 10.1080/10447318.2017.1278897. URL https://doi.org/10.1080/10447318.2017.1278897.

[69] R.-D. Vatavu and L.-B. Bilius. GestuRING: A Web-Based Tool for Designing Gesture Input with Rings, Ring-Like, and Ring-Ready Devices. In *ACM Symp. UIST*, pages 710–723. ACM, 2021.

[70] E. Velloso, D. Schmidt, J. Alexander, H. Gellersen, and A. Bulling. The feet in human–computer interaction: A survey of foot-based interaction. *ACM Comput. Surv.*, 48(2), sep 2015. ISSN 0360-0300. URL https://doi.org/10.1145/2816455.

[71] M. Verleysen and D. François. The Curse of Dimensionality in Data Mining and Time Series Prediction. In *Lecture Notes in Computer Science*, volume 3512, pages 758–770, 06 2005. ISBN 978-3-540-26208-4.

[72] J. Wang and F. Chuang. An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition. In *Trans. on Industrial Elec.*, 2012.

[73] Y. Wang and H. Ma. Real-Time Continuous Gesture Recognition with Wireless Wearable IMU Sensors. In *Int. Conf. on Healthcom*, 2018.

[74] H. Watanabe, M. Mochizuki, K. Murao, and N. Nishio. A Recognition Method for Continuous Gestures with an Accelerometer. In *UbiComp*. ACM, 2016.

[75] R. Xie and J. Cao. Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching. In *IEEE Sensors Journal*, 2016.

[76] L. Xu, K. Zhang, G. Yang, and J. Chu. Gesture recognition using dual-stream CNN based on fusion of sEMG energy kernel phase portrait and IMU amplitude image. *Biomedical Signal Processing and Control*, 2022.

[77] R. Xu, S. Zhou, and W. J. Li. MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition. In *IEEE Sensors Journal*, volume 12, pages 1166–1173, 2012.

[78] Y. Yan, C. Yu, X. Yi, and Y. Shi. Headgesture: Hands-free input approach leveraging head movements for hmd devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), dec 2018. URL https://doi.org/10.1145/3287076.

[79] J. Yim, E. Qiu, and T. C. N. Graham. Experience in the Design and Development of a Game Based on Head-Tracking Input. In *ACM Future Play*. ACM, 2008.

[80] T. Yu and H. Zhu. Hyper-Parameter Optimization: A Review of Algorithms and Applications. *ArXiv*, abs/2003.05689, 2020.

[81] Z. He, L. Jin, L. Zhen, and J. Huang. Gesture recognition based on 3D accelerometer for cell phones interaction. In *APCCAS*, 2008.

[82] D. Zhang, X. Wu, and C. Wang. Fine-Grained and Real-Time Gesture Recognition by Using IMU Sensors. In *ICPADS*, 2017.

[83] X. Zhang, X. Chen, W. Wang, J. Yang, V. Lantz, and K. Wang. Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors. In *IUI*. ACM, 2009.

[84] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. In *Transactions on Systems, Man, and Cybernetics - Systems*, volume 41(6), 2011.

[85] X. Zhang, J. Liu, J. Feng, Y. Liu, and Z. Ju. Effective Capture of Nongraspable Objects for Space Robots Using Geometric Cage Pairs. *IEEE/ASME Transactions on Mechatronics*, 25(1):95–107, 2020.

[86] Zhe Ji, Zhi-Yi Li, Peng Li, and MaoBo An. A new effective wearable hand gesture recognition algorithm with 3-axis accelerometer. In *Int. Conf. FSKD*, 2015.