Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 1, 133–147
DOI: 10.34768/amcs-2020-0011

A QUATERNION CLUSTERING FRAMEWORK

MICHAŁ PIÓREK [a,*], BARTOSZ JABŁOŃSKI [b]

[a] Department of Computer Engineering
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: michal.piorek@pwr.edu.pl

[b] Department of Control Systems and Mechatronics
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: bartosz.jablonski@pwr.edu.pl

Data clustering is one of the most popular methods of data mining and cluster analysis. The goal of clustering algorithms is to partition a data set into a specific number of clusters for compressing or summarizing original values. There are a variety of clustering algorithms available in the related literature. However, the research on the clustering of data parametrized by unit quaternions, which are commonly used to represent 3D rotations, is limited. In this paper we present a quaternion clustering methodology including an algorithm proposal for quaternion based k-means along with quaternion clustering quality measures provided by an enhancement of known indices and an automated procedure of optimal cluster number selection. The validity of the proposed framework has been tested in experiments performed on generated and real data, including human gait sequences recorded using a motion capture technique.

Keywords: data clustering, quaternions data processing, human gait data processing.

1. Introduction

Data clustering is an extensively explored area in the field of data mining. Roughly speaking, the problem is to group a given set of data points into a fixed number of clusters in such a way that points within the cluster are similar and those from two different clusters are dissimilar. In most cases the clustering operation is based on attributes of investigated objects. The distance function is also frequently used as a similarity dissimilarity measure. A very good reference on clustering concepts is provided by Han *et al*. (2011). A survey of clustering data mining techniques can be found in the work of Berkhin (2006) or Liao (2005) for a time series case.

Clustering has applications in many fields, e.g., image processing (Wu and Leahy, 1993; Bandyopadhyay and Maulik, 2002; Koster and Spann, 2000), sociology (Cantador and Castells, 2006; Austin *et al*., 2005), medicine (Himberg *et al*., 2004), traffic analysis (Reumerman *et al*., 2005), criminology (Grubesic, 2006) or space science (Gariel *et al*., 2011). In each case, different clustering concepts are used. Very often the choice of the method is tightly coupled with the investigated domain. We are primarily interested here in the gait kinematic data domain, especially unit quaternion parametrization of kinematic data. Within the field of quaternion clustering techniques, one can observe limited research.

There are many clustering algorithms available in the literature. The best-known clustering methods include hierarchical methods, partitioning methods, density-based methods and grid-based methods. Very often an even more detailed breakdown is available, but for the purpose of this paper there is no need to provide a comprehensive categorization of the existing methods, only to have a clear understanding of the underlying subject.

Specifically, partitioning methods are in the scope of the presented research. These methods for the input set of n objects partition those into k-groups such that each group must contain at least one object and each object is assigned to only one group. Usually, partitioning criteria

*Corresponding author

are a distance measure. It is also worth emphasizing that very often these methods use an iterative relocation technique. This means that it moves already assigned objects between the clusters with the goal of finding a better assignment. One of the most popular partitioning methods is the k-means algorithm, widely used in scientific and industrial applications. The popularity of the method is caused by its relative simplicity of understanding, implementation and application to the requested domain.

The main goal of this paper is to present a quaternion clustering methodology. The approach is an outcome of analysis of chaotic properties of processes described by quaternions presented in the work of Piórek (2018), where the initial idea of quaternion clustering appeared. Here this idea is enriched and wrapped up into a framework including the k-means algorithm generalization to the quaternion domain, a generalized set of methods to investigate clustering results, an automated procedure for estimation of the number of clusters and a test data generation framework.

The proposed methodology was tested on generated data and human gait data obtained from the Center for Research and Development of the Polish-Japanese Academy of Information Technology (`www.bytom.pja.edu.pl`). Based on the observation that human gait rotations can be grouped into several primitives, the presented approach can group the set of rotations from a single body bone separately. The numerical tests of the method were performed on several bones: tibia, femurs and feet. In this sense this approach can be called bone-wise data clustering since it is performed on data recorded from each body part separately.

The paper is organized in the following way. In Section 2 the state of the art for clustering in the quaternion domain is presented. Section 3 describes quaternion data parametrization details. In Section 4 the original definition of the k-means algorithms is recalled, along with the proposal to generalize the algorithm for quaternion parametrization. Section 5 provides quaternion clustering quality measures. Section 6 defines the test data generation framework. The numerical results and conclusions are discussed in Sections 7 and 8 respectively.

2. State of the art

Clustering techniques are a widely explored field of research. Despite the abundance of generic clustering methods (surveyed, e.g., by Berkhin (2006)), there is a limited amount of research performed for clustering algorithms adapted fully to the quaternion domain—well suited for rotation clustering. As methods fully adapted to the quaternions domain we treat those where all operations in the clustering algorithm are based on quaternion algebra and quaternion properties, e.g., the distance function is the quaternion distance function, initial random cluster center generation is random quaternion generation, and selection of new cluster centers is performed based on the quaternion properties.

In the paper by Shi and Funt (2007), a quaternion clustering application is used for quaternion color segmentation. In this approach the quaternion mean is applied in a modified k-means algorithm, although the distance function is not a quaternion metric. This is natural since quaternion feature vectors used there represent the RGB color and may consist of quaternions which are not necessarily unit quaternions. As a consequence, this approach is not suitable for rotation clustering.

Another approach to partial clustering in the quaternion domain is by using face landmark labeling in 3D (Creusot *et al.*, 2010). The distance function applied here is indeed a quaternion distance; however, it seems that new cluster centers and initial cluster centers are not computed based on quaternion properties.

A kernel-based quaternion k-means application has been proposed for aligned cluster analysis for temporal segmentation of human motion by Zhou *et al.* (2008). In this case the similarity matrix used in the clustering process is not computed using a quaternion metric.

In the paper of Risojević and Babić (2013), k-means clustering is used in unsupervised learning of quaternion features for image classification, though the quaternion vectors are not unit quaternions and are clustered as 4D vectors.

The reason for research on quaternion clustering is to propose an algorithm based only on quaternion algebra and quaternion properties. All the previously mentioned approaches seem at some point to step out from the quaternion domain and use operations not based on quaternion properties. In many cases the proposed algorithms even use the quaternion domain distance measure, but a new cluster generation or a cluster assignment is not based on the quaternions domain. Sometimes, even though the input data are represented in the quaternion form, in the clustering process they are treated as a 4-dimensional vector and the classic version of k-means is used.

To the best of our knowledge, measures of clustering quality were not previously generalized and analyzed for quaternion parametrization, so that both the clustering algorithm and the clustering quality assessment could be performed in the quaternion domain. This was the second motivation for the proposed approach.

It is also worth emphasizing that application of quaternions to color representation in image processing is also very commonly used (e.g., Pei and Cheng, 1999; Shi, 2005), which in the presence of the quaternions clustering technique can open new fields of research.

3. Data parametrization

Recently quaternions have been gaining importance and many practical applications could be observed. This includes, but is not limited to, areas such as aerospace or underwater attitude control (Chaturvedi *et al.*, 2011), motion capture data processing (Jabłoński, 2011), color image transformation (Ell and Sangwine, 2007) and many others.

Quaternions are hyper-complex numbers defined as $Q = w + \mathbf{v}$, where w is called the real part (scalar) and $\mathbf{v} \in \mathbb{R}^3$ is the imaginary part. One can find a detailed description and analysis of quaternion properties in the work of Johnson (2003).

Specifically, unit quaternions are a commonly used parametrization for 3D rotations. Unit quaternions are a subset of the quaternion group whose magnitude equals one,

$$||q|| = \sqrt{qq^*} = \sqrt{w_1^2 + \mathbf{v} \cdot \mathbf{v}} = 1. \tag{1}$$

Assuming that q_r defines a vector in the 3D space, one can calculate a rotated vector by quaternion multiplication of

$$q_r' = qq_rq^{-1}, \tag{2}$$

where q defines the rotation angle and axis.

Datasets taken into consideration in this paper concern some type of parametrization of rotations. All 3D rotations form $SO(3)$, which is a special orthogonal group. Quaternions can be considered one parametrization of this group.

For the discussed clustering algorithm, it is crucial to have a well-defined distance function (metric) between two elements of a dataset. Any distance function d used for handling rotation data needs to satisfy both usual axioms for metrics and $SO(3)$ properties (Huynh, 2009):

- $d(R_1, R_2) = 0 \leftrightarrow R_1 = R_2$,
- $d(R_1, R_2) = d(R_2, R_1)$,
- $d(R_1, R_3) \leq d(R_1, R_2) + d(R_2, R_3)$,
- function d preserves the topology of $SO(3)$,
- function d is left/right invariant or bi-invariant.

Huynh (2009) provides a survey for various distance functions for the $SO(3)$ group and specifically quaternions. Let us summarize the formulas collected therein:

- Euclidean distance between angles,

$$d_1((\alpha_1, \beta_1, \gamma_1), (\alpha_2, \beta_2, \gamma_2)) \\ = \sqrt{d(\alpha_1, \alpha_2)^2 + d(\beta_1, \beta_2)^2 + d(\gamma_1, \gamma_2)^2}; \tag{3}$$

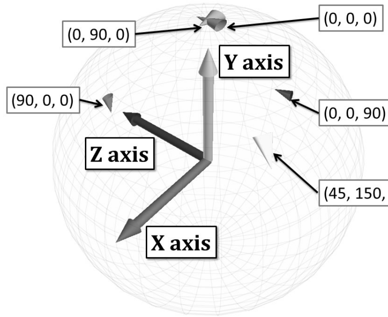


Fig. 1. Sphere of rotation to visualize rotational data sets (Jabłoński, 2008a).

- difference of quaternions,

$$d_2(q_1, q_2) = \min\{||q_1 - q_2||, ||q_1 + q_2||\}; \tag{4}$$

- inner product,

$$d_3(q_1, q_2) = \arccos(|q_1 \cdot q_2|); \tag{5}$$

- inner product without inverse cosine,

$$d_4(q_1, q_2) = 1 - |q_1 \cdot q_2|; \tag{6}$$

- deviation from the identity matrix,

$$d_5(R_1, R_2) = ||I - R_1R_2^T||_F; \tag{7}$$

- geodesic on the unit sphere,

$$d_6(R_1, R_2) = ||\log(R_1R_2^T)||. \tag{8}$$

Distance measure d_6 is functionally equivalent to d_3 and has all the required properties (metric axioms, bi-variance, respecting the $SO(3)$ topology). It was used in some of our previous research (Jabłoński, 2008a; 2011) and provided very good results in practical applications. It can be interpreted as the length of the geodesic curve on the quaternion unit sphere. It also denotes the amount of energy or rotation needed to rotate quaternion q_1 to the rotation defined by quaternion q_2.

In the following presentation of results, we are going to visualize the rotational data using the concept previously proposed by Jabłoński (2012; 2008b) called the sphere of rotations; see Fig. 1.

The visualization is independent from the parametrization for the rotational space. Specifically, it is valid for quaternions, which is the default parametrization used throughout the paper. In all experiments presented here, we are dealing with data sets consisting of multiple rotational data points.

In the selected approach (presented in Fig. 1), there are three axes in the figure—each corresponds to one

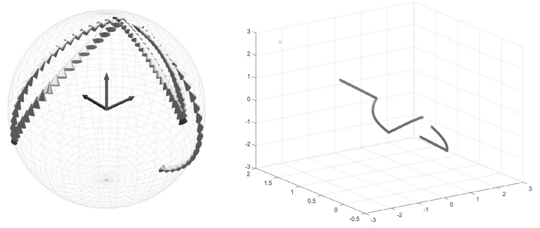


Fig. 2. Comparison of a sample data set visualized using the sphere of rotations (left) and raw Euler angles (right).

operation of rotation. All data items are represented as the directed arrow located on the sphere. Its purpose is to visualize the rotational item in comparison with the reference orientation presented as a gray item. This could be visualised by any object, which allows us to distinguish the direction (hence the arrow instead of just a ball). For each data item, its orientation is parametrized as three consecutive rotations around the axes described by Euler angles (x, y, z). There are three basic examples in the figure visualized as arrows of different shades with the corresponding rotations: medium gray for $(90, 0, 0)$, light gray for $(0, 90, 0)$ and dark gray for $(0, 0, 90)$. The white-gray arrow represents the data point of rotations $(45, 150, 0)$. In some cases we may need to present a large amount of data. In such a case, some elements of a data set can be presented as dots instead of arrows to improve readability.

In Fig. 2 one can observe the difference and visualization properties of the sphere of rotation (left) versus the raw Euler angles plot (right). In both the cases the same set of rotational data points were visualized (note that for the left figure only every 5 items are displayed for better visibility). As one can observe, the sphere of rotations preserves the rotational character of the underlying data model, including the spatial relationship and the distance between elements. This will help us in visualizing the relationship between data items in clusters in a more direct way.

4. Quaternion k-means algorithm

The classic version of the k-means algorithm is the main example of partitioning methods of data clustering algorithms to classify an input data set through a fixed (known *a priori*) number of clusters k. Assume that we have an input data set D which contains n objects in a Euclidean space. The k-means algorithm, by partitioning, distributes all objects from the input data set into k clusters $C_1, C_2, \ldots, C_k$ in such a way that $C_i \subset D$ and $C_i \cap C_j = \emptyset$ for $1 \leq i, j \leq k$. Each of the clusters is represented by a centroid—the central point of the cluster. The centroid is calculated as the mean of all objects assigned to this cluster. For cluster assignment, an objective criterion is used. Typically, it is one of the distance measures. The algorithm is an iterative procedure; in each step it attempts to find new clusters with the goal of obtaining a better assignment. Once the choice of the cluster center is stabilized, the algorithm ends (Han *et al.*, 2011).

We can summarize the algorithm steps in the following way:

1. Randomly choose k points in a given space. These points represent the centers of the clusters (centroids).

2. Assign each point from the input set to the closest centroid based on the distance measure.

3. Once all objects are labeled, recalculate the position of the centers.

4. Repeat Steps 2 and 3 until the positions of the centers change no longer.

This form of the k-means approach is not sufficient for gait kinematic quaternion data clustering. Several enhancements need to be incorporated to apply the algorithm in the mentioned domain. Therefore, we propose a generalization of k-means clustering for the quaternion domain.

Quaternion time series can describe a particular body rotation in time (e.g., human gait recordings captured on a treadmill). While observing (but not only human) gait from each bone perspective in time, we can observe that some rotations appear periodically. This is because during normal walking we repeat some movements. This hint leads to checking whether we can group bone rotations into a fixed number of groups.

We take into consideration the unit quaternion data set created from samples of unit quaternions time series:

$$\begin{aligned} Q(n) &= (q_1, q_2, \ldots, q_N) \\ &= (w_1 + ix_1 + jy_1 + kz_1, \\ &\quad w_2 + ix_2 + jy_2 + kz_2, \ldots, \\ &\quad w_N + ix_N + jy_N + kz_N). \end{aligned} \tag{9}$$

We have to ensure that clustering results remain in the quaternion domain. This requires several extensions to be introduced. In the first step, examine a random generation of initial clusters centers. To keep the physical sense of rotational data, which are the input set of the clustering, the randomly generated cluster center has to fulfill a couple of requirements:

- The input data set consists of rotations described by unit quaternions only, hence the randomly generated cluster center has to be a unit quaternion as well (it should be taken from the unit quaternion sphere).

- Since we consider rotational data captured from real systems (e.g., a particular part of the skeleton), we have to bear in mind that the rotated body has its own range of available rotations.

- We assume the above-mentioned range will be defined as a rotation interval $[q_a, q_b]$, where q_a and q_b are minimal and maximal rotations, respectively.

- As the minimal rotation q_a we treat the rotation described by the quaternion with a minimal quaternion's angle (similarly for maximal).

$$q_a = \arg\min_q 2\arccos(\mathrm{Re}(q)), \tag{10}$$

$$q_b = \arg\max_q 2\arccos(\mathrm{Re}(q)). \tag{11}$$

Consequently, we initialize k centroids by random selection from the existing input data set based on a uniformly distributed index (the position of the quaternion in the data set).

The second area of k-means application to the quaternion and rotation domain is cluster assignment. The classic k-means approach assumes the Euclidean distance measure as an objective function while labelling the data set into clusters. In the case of a unit quaternion, treating a quaternion as a 4D-vector loses its rotational sense. In the light of this fact, we propose to use one of distance measures from the quaternion domain. In the proposed algorithm the distance d_6 described in the previous section is used. The distance function can be then interpreted as the minimum length of a geodesic line connecting two quaternions in the four-dimensional unit sphere. After some transformations we obtain a simplified formula:

$$d(q_1, q_2) = 2\arccos(\mathrm{Re}(q_1^* q_2)). \tag{12}$$

Finally, to each rotation in the input data set we assign the cluster C_i (where $i = 1, \ldots, k$), which minimizes the distance between the center of the cluster and the rotation.

Once all input rotations in one iteration of the k-means algorithm are assigned to one of the clusters, we have to find a new center of each cluster, which should improve the clustering quality. In the classic version of the k-means algorithm, the new center for the cluster is calculated as an average of all samples from the input data set belonging to this cluster. For quaternion domain clustering, the averaging method should be taken from this domain as well. Quaternion averaging is a subject of many publications. A survey of the existing methods can be found in the work of Markley *et al.* (2007). For the purposes of this research, algebraical quaternion averaging has been selected. A new quaternion cluster center is the algebraic mean of all samples belonging to the particular cluster. Its worth emphasizing that the choice of the method might be a crucial factor affecting the clustering quality evaluation (as well as the choice of the method of random initialization and the distance function), which can be a direction of further research.

5. Generalization of clustering quality measures for quaternion data

In the paper by Maulik and Bandyopadhyay (2002), one can find performance evaluation of clustering algorithms based on the analysis of selected validity indices: the Davies–Bouldin index (DB) (Davies and Bouldin, 1979), Dunn's index (DI) (Dunn, 1974), and the Calinski–Harabasz index (CH) (Caliński and Harabasz, 1974). In the following experiments we will employ these indices to analyze properties of the proposed clustering algorithm for quaternions.

Based on the compact description by Maulik and Bandyopadhyay (2002) and appropriate references, we propose the following measures for quaternion clustering indices:

- QDB index (generalized Davies–Bouldin index),

- QDI index (Dunn's generalized index),

- QCH index (generalized Calinski–Harabasz index).

All of the proposed indices are adjusted to work in the quaternion domain. They are based on distance measures and hence to apply them to the quaternion domain, the appropriate quaternions distance measure should be used. We propose to use the distance measure between two rotations parametrized by unit quaternions, described in Table 1 as d_6 with its simplified formula described in Eqn. (12).

In the case of the Davis–Bouldin index, the scatter within the i-th cluster is used. For the quaternion's domain we propose to use distance d_6, which results in the following formula:

$$\begin{aligned} S_i &= \frac{1}{|C_i|} \sum_{x \in C_i} \{d(x, z_i)\} \\ &= \frac{1}{|C_i|} \sum_{x \in C_i} \{2\arccos(\mathrm{Re}(x^* z_i))\}. \end{aligned} \tag{13}$$

The distance between quaternion clusters C_i and C_j is defined using the proposed quaternion distance:

$$d_{i,j} = d(z_i, z_j) = 2\arccos(\mathrm{Re}(z_i^* z_j)), \tag{14}$$

where z_i denotes the i-th quaternion cluster center. The final form of the Davies–Bouldin index formula generalized for quaternions can be denoted as

$$\mathrm{QDB} = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt}, \tag{15}$$

where

$$R_{i,qt} = \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}.$$

Dunn's index also requires significant changes to enhance it and use during quaternion clustering index validation. First of all, the computation of the diameter of the quaternion set and of the distance between two quaternion sets has to be incorporated. Suppose that S and T are nonempty subsets of $\mathbb{H}$.

The diameter Δ of S is defined as

$$\begin{aligned} \Delta(S) &= \max_{x,y \in S} \{d(x, y)\} \\ &= \max_{x,y \in S} \{2 \arccos(\mathrm{Re}(x^* y))\}. \end{aligned} \tag{16}$$

The distance δ between two quaternion sets S and T can be defined using a quaternion distance measure:

$$\begin{aligned} \delta(S) &= \min_{x \in S, y \in T} \{d(x, y)\} \\ &= \min_{x \in S, y \in T} \{2 \arccos(\mathrm{Re}(x^* y))\} \end{aligned} \tag{17}$$

As a result, Dunn's index for quaternion clustering can be computed using the original formula:

$$\mathrm{QDI} = \min_{1 \leq i \leq K} \left\{ \min_{1 \leq j \leq K j \neq i} \left\{ \frac{\delta(C_i, C_j)}{\max\limits_{1 \leq k \leq K} \{\Delta(C_k)\}} \right\} \right\}. \tag{18}$$

The last of the listed indices, the Calinski–Harabasz one, is computed based on the traces of two matrices: the between-clusters scatter matrix B and the within-clusters scatter matrix W. The following formula describes the CH index:

$$CH = \frac{[\mathrm{trace}\, B/(K-1)]}{[\mathrm{trace}\, W/(n-K)]}, \tag{19}$$

where

$$\mathrm{trace}\, B = \sum_{k=1}^{K} n_k \left\| z_k - z \right\|^2. \tag{20}$$

and

$$\mathrm{trace}\, W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_i - z_k \right\|^2 \tag{21}$$

To compute the same traces in the quaternion domain, we propose to replace the Euclidean distance with the respective distance function from the quaternion domain:

$$\mathrm{trace}\, B = \sum_{k=1}^{K} n_k (2 \arccos(\mathrm{Re}(z_k^* z)))^2 \tag{22}$$

and

$$\mathrm{trace}\, W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} 2 \arccos(\mathrm{Re}(x_i^* z_k))^2. \tag{23}$$

Finally, the formula of the proposed QCH index can be written as

$$\mathrm{QCH} = \frac{\dfrac{1}{K-1} \sum\limits_{k=1}^{K} n_k (2\arccos(\mathrm{Re}(z_k^* z)))^2}{\dfrac{1}{n-K} \sum\limits_{k=1}^{K} \sum\limits_{i=1}^{n_k} 2\arccos(\mathrm{Re}(x_i^* z_k))^2}. \tag{24}$$

The objectives are to minimize the QDB index, maximize the QDI index and maximize the QCH index to establish proper clustering.

5.1. Multi-index selection algorithm for an expected number of clusters. One of the tasks of clustering quality indices is to select an optimal number of clusters for the given data set. This is possible by finding the minimum or maximum of index values.

We observed that, in many practical cases of quaternion clustering, these measures do not always have a well-defined global optimum. Depending on data diversity, some indices may not possess any extrema. For any kind of automated clustering tasks, we must to have an algorithm for selection of the optimal number of clusters based on given measures. Hence, we propose to construct a heuristic measure of this kind.

We propose to base the selection algorithm on a multi-index local extrema analysis.

Algorithm 1. Selecting the expected number of clusters.

Input: QDB, QDI, QCH arrays $1, \ldots, k$
Output: K_E—expected number of clusters
Initialization: A—accumulator array $1, \ldots, K$

```
for i = 2 to K − 1 do
    if QDB[i−1] > QDB[i] and QDB[i+1] > QDB[i] then
        A[i-1]++
        A[i]++
        A[i+1]++
    end if
    if QDI[i − 1] < QDI[i] and QDI[i + 1] < QDI[i] then
        A[i-1]++
        A[i]++
        A[i+1]++
    end if
    if QCH[i−1] < QCH[i] and QCH[i+1] < QCH[i] then
        A[i-1]++
        A[i]++
        A[i+1]++
    end if
end for
return K_E arg max_i A[i]
```

This kind of routine is based on a multi-criteria voting mechanism for the optimal number of clusters. Initial experiments indicate that using a relaxed version of the method (in which each local optimum votes for its and neighboring indices) can lead to more reliable results. In the case of observation of the cumulative effect of lines 2, 7 and 12, one should select the middle value (median) as an optimal clusters number. It is also possible to use a strict voting mechanism, in which each optimum votes only for its own index.

The returned value K_E is not guaranteed to be an optimal number of clusters. Nevertheless, this heuristic approach yields a good indication of the automatic cluster selection approach based on multi-index criteria. In the following numerical analysis, we will use it to analyze the properties of the proposed clustering algorithm.

6. Test data preparation

To be able to test the properties of the proposed approach, we have to prepare test data sets. Since there are no well established test data sets for quaternions, we propose to use the following test data generation approach.

In each case, the elementary task was to generate two or more clusters of quaternions. Each cluster consisted of a defined number of samples generated from normal distribution. The parameters for a generation single cluster set was the following family: the cluster center, the number of samples, and the distribution variance.

The goal of data preparation is to obtain a data set containing the number of randomly distributed clusters of quaternions. There are several ways to define quaternion random variables (Loots *et al.*, 2013). We will follow the wrapping approach discussed by Johnson (2003).

The multivariate non-degenerate normal distribution in k-dimensional case is defined as

$$f_x(x,\mu) = \frac{1}{\sqrt{(2\pi)^k |K|}} \exp\Big(-\frac{1}{2}(x-\mu)^T K^{-1}(x-\mu)\Big), \tag{25}$$

where K is a positive definite symmetric covariance matrix.

The wrapping approach assumes that the random quaternion element is calculated as a regular zero mean multivariate normal distribution in tangent space TS^3 rotated at selected q_m quaternion representing the distribution mean. The approach is presented in Fig. 3.

Having a simplified model and making use of quaternion properties, we obtain the following formula for the probability density function:

$$p(q) = c\exp\Big(-\frac{1}{2}\log(q_M\cdot q)^T K^{-1}\log(q_M\cdot q)\Big), \tag{26}$$

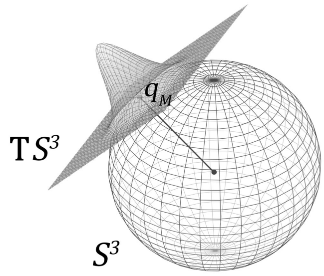


Fig. 3. Visualization of the probability distribution function for quaternions (Jabłoński, 2008b).

where c is a normalization parameter (in this case, equal to 1) and K is the covariance matrix. In the following experiments, we will make use of a simplified random model with fixed variance at value σ_q for all dimensions.

Several groups of data sets have been generated:

Data set 1. The data set Q_{nC} consists of n clusters, each containing 1000 elements generated from a Gaussian distribution with variance 0.005 and a defined quaternion mean. Each set is described by the list of clusters centres described by three Euler angles (r_x, r_y, r_z):

$$\begin{aligned} Q_{2C} &= \{(0,0,30);(0,0,60)\},\\ Q_{3C} &= \{(0,0,30);(0,0,60);(-30,0,60)\},\\ Q_{4C} &= \{(0,0,30);(0,0,60);(-30,0,60);\\ &\quad (30,0,60)\},\\ Q_{5C} &= \{(0,0,30);(0,0,60);(-30,0,60);\\ &\quad (30,0,60);(0,0,90)\},\\ Q_{7C} &= \{(0,0,30);(0,0,70);(-50,0,60);\\ &\quad (30,0,70);(0,0,110);(0,0,-30)\},\\ Q_{10C} &= \{(0,0,30);(0,0,60);(-30,0,60);\\ &\quad (30,0,60);(0,0,90);(0,0,-30);\\ &\quad (0,0,-60);(-30,0,-60);(30,0,-60);\\ &\quad (0,0,-90)\} \end{aligned} \tag{27}$$

One can observe visualization of a sample data set Q_{5C} in Fig. 4. Each cluster is marked with a different shade. See Fig. 1 for convention of the sphere of rotation.

Data set 2. Overlapping clusters: $Q_{n\text{Dist}}$ consists of 2 clusters, each containing 1000 elements generated from a Gaussian distribution with variance 0.005 and the angle distance of cluster centres nDist for a selected axis. Each set is described by the list of clusters centres described by 3 angles (r_x, r_y, r_z):

$$\begin{aligned} Q_{10\text{Dist}} &= \{(0,0,30);(0,0,40)\},\\ Q_{20\text{Dist}} &= \{(0,0,30);(0,0,50)\},\\ Q_{30\text{Dist}} &= \{(0,0,30);(0,0,60)\},\\ Q_{40\text{Dist}} &= \{(0,0,30);(0,0,70)\},\\ Q_{50\text{Dist}} &= \{(0,0,30);(0,0,80)\}, \end{aligned}$$

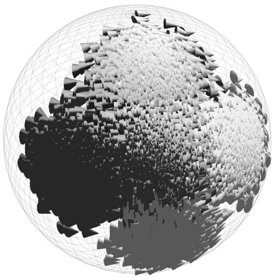
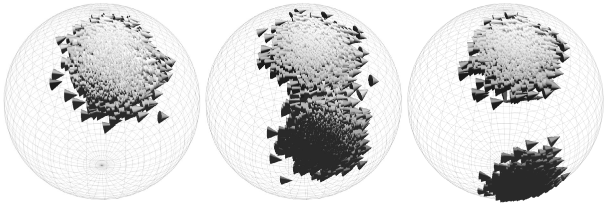
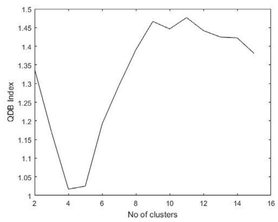


Fig. 4. Visualization of the test dataset called Q_{5C} (Piórek, 2018).

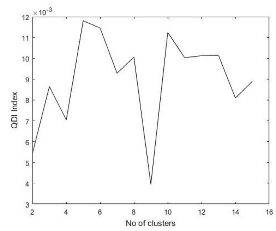


(a) $Q_{10\text{Dist}}$ (b) $Q_{40\text{Dist}}$ (c) $Q_{70\text{Dist}}$

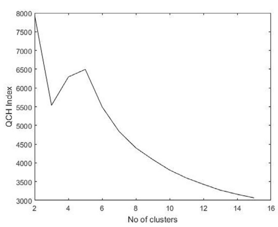
Fig. 5. Visualization of three datasets with a variable amount of overlapping clusters (Piórek, 2018).



(a) QDB index



(b) QDI index



(c) QCH index

Fig. 6. Procedure of finding an optimal number of clusters for data generated around 5 clusters.

$$Q_{60\text{Dist}} = \{(0, 0, 30); (0, 0, 90)\},$$
$$Q_{70\text{Dist}} = \{(0, 0, 30); (0, 0, 100)\}.$$

Having these kinds of data sets available for the experiment, it is possible to reconstruct the parameters of normal distribution of the given data set using maximum likelihood estimators for quaternions described by Markley *et al*. (2007). However, in the present paper, we will limit the usage of these methods to find an average as described in Section 5.

7. Numerical results

Three kinds of experiments were performed to test the proposed quaternion's domain clustering k-means algorithm:

1. Finding optimal clustering for the data generated with the proposed quaternion's test data generation framework.
2. Finding optimal clustering for the generated data with different degrees of data overlapping.
3. Finding optimal clustering for live recorded real data.

Each experiment consists of two stages:

- Testing the proposed quaternion's clustering quality measures against the number of clusters with the aim of minimizing QDB, maximizing QDI and QCH.
- Performing the clustering for the selected number of clusters and visualizing the results.

In the first stage the procedure was to iterate through the number of cluster values from $Cn_{\min}$ to $Cn_{\max}$ (here arbitrarily selected as $Cn_{\min} = 2$ and $Cn_{\max} = 15$), repeat the clustering $N_{\text{rep}} = 3$ times and averaging each of the quaternion's clustering quality measure (QDB, QDI and QCH). In consequence we obtain three dependencies: averaged QDB, QDI and QCH with respect to the number of clusters. Figure 6 illustrates the procedure established for the calculation of the number of clusters for data spread out around 5 clusters.

Calculations of the same set of metrics for ground truth original clustering data sets resulted in $\text{QDB}_{\text{original}} = 2.0678$, $\text{QDI}_{\text{original}} = 0.00051$, $\text{QCH}_{\text{original}} = 1118.37$. These values are of the same magnitude as the ones obtained with k-means clustering. However, the absolute values differ significantly from those calculated after the clustering procedure. After a quick analysis of the original cluster positions (see Fig. 5), we conclude that it is a result of the amount of data points overlapping between clusters.

Once these dependencies were computed, the multi-index selection algorithm described in Section 5.1 was run to obtain a valid number of clusters.

In the second stage of the procedure, clustering was performed again for the numbers of clusters selected in the previous stage. Several results were selected to visualize the results using the sphere of rotations. We did not visualize every performed test case, but only those which are a good basis for the analysis of the results as regards comparison of the similarity or dissimilarity of the original ground truth clusters for the given datasets.

7.1. Testing clustering for a varying number of clusters and randomly generated data. The first investigation was performed on 6 data sets used in the test framework described in Section 6. The data were generated around 2, 3, 4, 5, 7 and 10 cluster centres, respectively, each containing 1000 elements generated from a Gaussian distribution with variance 0.005 and a specified quaternion mean. The calculation of the number of clusters with respect to the quality measure used for the generated data is summarized in Table 1 (using the algorithm proposed in Section 5.1).

Figures 7, 8 and 9 present the k-means clustering results for the four data sets (containing 2, 3, 4 and 5 generated clusters) for the selected number of clusters collected in Table 1.

It is interesting to see how the clusters align with the original centres of random generation of quaternions in the quaternion sphere. The comparison of the clustered data against the originally generated case for 3 clusters is presented in Fig. 9.

In all performed experiments the criteria of selecting an optimal number of clusters were to locally minimize or maximize all of the quaternion validity indices at the same time. The actual calculation is based on the algorithm described in Section 5.1. Apparently, this is more suitable than selection based on global suprema of quality measures dependencies. In Fig. 10 the number of clusters for the data generated around 7 clusters is depicted. Calculations of the same set of metrics for ground truth original clustering data sets resulted in $\mathrm{QDB}_{\mathrm{original}} = 2.8798$, $\mathrm{QDI}_{\mathrm{original}} = 0.0695$, $\mathrm{QCH}_{\mathrm{original}} = 1265.45$. Similarly to the case of five clusters, also these values are difficult to compare with the results obtained for k-means clusters.

Selecting the number of clusters based on the

Table 1. Calculated number of clusters obtained from multi-index criteria (see Algorithm 1, Section 5.1).

Dataset	No. of clusters
QCluster02C1000S0.005v.mat	2
QCluster03C1000S0.005v.mat	3
QCluster04C1000S0.005v.mat	3
QCluster05C1000S0.005v.mat	4
QCluster07C1000S0.005v.mat	8
QCluster10C1000S0.005v.mat	8

(a) 2 groups of generated data (b) 3 groups of generated data

Fig. 7. Clustered rotations for the generated data set (2 and 3 groups).

(a) 4 groups of generated data (b) 5 groups of generated data

Fig. 8. Clustered rotations for the generated data set (4 and 5 groups).

(a) generated data (b) clustered data

Fig. 9. Generated data versus the same data clustered.

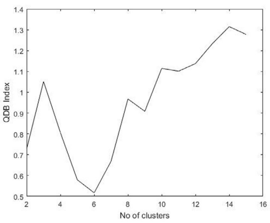
global extrema provides two clusters as a valid number, while selection based on the proposed algorithm provides 8 clusters, which complies more with reality. The clusterings for both possible numbers of clusters are visualized in Fig. 11.

7.2. Testing quaternion k-means clustering against Euler angles baseline k-means clustering. As a part of this experiment block, one more exercise was performed. To address advantages of the proposed quaternion k-means clustering approach, its results were compared against the baseline k-means algorithm clustering for the same rotations given in Euler angle parametrization. We repeated this test for all previously mentioned cluster numbers and observed that with a growing cluster number, quaternion-based k-means seems to perform more promisingly. It can be nicely depicted by clustering 7 or 10 clusters data sets. Figures 12 and 13 present rotation clustering into 7 and 10 clusters for data sets containing 7 and 10 clusters, respectively.

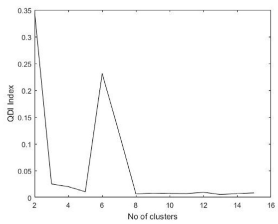
In the case of quaternion k-means, the computed clusters seem to be separated better from each other than in the case of Euler angles baseline k-means. In the figures presenting results for Euler angles clustering several areas with low separability between cluster are highlighted

Table 2. Number of clusters calculated using multi-index criteria for different data overlapping degrees.

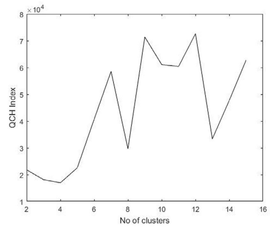
Dataset	No. of clusters
`QCluster2C1000S0.005v_10dist.mat`	2
`QCluster2C1000S0.005v20dist.mat`	2
`QCluster2C1000S0.005v30dist.mat`	2
`QCluster2C1000S0.005v40dist.mat`	2
`QCluster2C1000S0.005v50dist.mat`	3
`QCluster2C1000S0.005v60dist.mat`	3
`QCluster2C1000S0.005v70dist.mat`	3



(a) QDB index

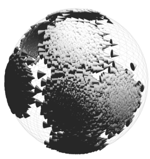


(b) QDI index

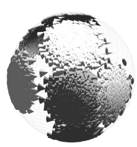


(c) QCH index

Fig. 10. Procedure of finding the number of clusters for data generated around 7 clusters.



(a) 2 clusters



(b) 7 clusters

Fig. 11. Clustering results for two differently estimated numbers of clusters.

(circle lines). The same effect was not observed in the case of quaternions k-means. To confirm this property one would need to perform more numerical experiments and compare cluster separability using some measure. This can be a subject of further research.

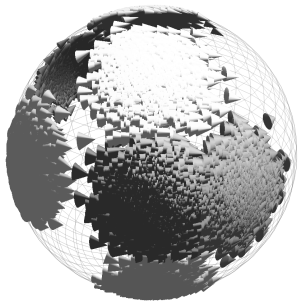
7.3. Testing clustering for a varying degree of data overlapping. In the second experiment the performance of the proposed algorithm was tested against various degrees of data overlapping. The experiment was carried out on data sets generated around 2 clusters with different amounts of overlapping, from 10 to 70 degrees.

The main goal of this experiment is to see how valid the proposed algorithm is with the increased complexity of the input data set and what is the biggest amount of overlapping when the proposed quaternion clustering quality measures provide the real number of clusters of underlying data. The calculation of the number of clusters for the generated data with respect to different degrees of data overlapping is collected in Table 2.

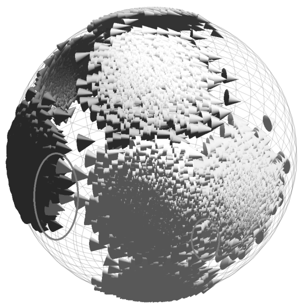
7.4. Testing clustering for recorded real gait kinematic data. The proposed method and clustering validity indices were also tested on gait kinematic data, which were recorded in the Human Motion Laboratory (HML) of the Polish-Japanese Academy of Information Technology. The recordings were taken using the Vicon Motion Kinematics Acquisition and Analysis system equipped with 10 NIR (near infrared) cameras registering movements of a subject wearing a suit with attached markers. The whole process is called the motion capture process.

Three bones from the human kinetic chain were recorded: femurs, tibia and feet. The calculated number of clusters for all bones is gathered in Table 3.

As highlighted in the previous testing, we employed the algorithm described in Section 5.1 for selecting the number of clusters, which is based on the local maxima of the quality measures dependencies that can provide a more reasonable number of clusters than using global ones.

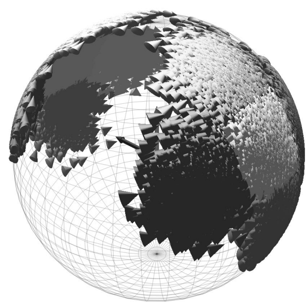


(a) quaternion k-means

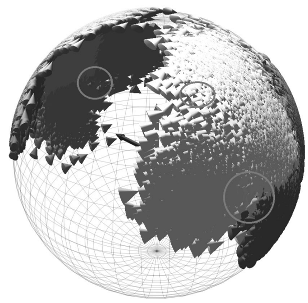


(b) Euler angles k-means

Fig. 12. Quaternion vs. Euler angles k-means: 7 clusters.



(a) quaternion k-means



(b) Euler angles k-means

Fig. 13. Quaternion vs. Euler angles k-means: 10 clusters.

The clustering results for the respective number of clusters from Table 3 for all available bones are presented in Figs. 16, 17 and 18.

7.5. Applying gait data clustering for automatic gait phase recognition. According to Abhayasinghe and Murray (2014), the human stride cycle can be further divided into sub-phases. Based on the clustering results and the stride analysis, we propose to use clustering to detect the stride phase.

In the next experiment we relied on the results obtained for a data set consisting of 921 samples of the right femur motion of one subject, which was automatically clustered into 3 classes (see Fig. 18).

Table 3. Number of clusters calculated using multi-index criteria for live gait kinematic data.

Dataset	No. of clusters
`lfemur.mat`	9
`lfoot.mat`	2
`ltibia.mat`	12
`rfemur.mat`	3
`rfoot.mat`	2
`rtibia.mat`	3

Table 4 presents the list of subphases, their average end of the interval (relative to the whole stride length) and the classification criteria based on clustering results for 3 classes.

Based on the feature points of the gait cycle (local minima and maxima) defined by Abhayasinghe and Murray (2014), the subphases described in Table 4 were identified for gait kinematic data visualized in Fig. 19. For clarity, only the angle component of the femur motion was used. Additionally, clusters computed by the method introduced in Section 4 were marked using the following convention: Cluster 1—light grey, Cluster 2—medium grey, Cluster 3—dark grey.

Based on the clustering results obtained in Section 7.4 we compared automatic stride phase recognition accuracy against ground truth phases labelling performed manually. The results are presented in Table 5.

Automatically clustered data with the appropriate classification criteria lead to detection of gait sub-phases with high accuracy. This is a solid basis for future research and numerous practical applications including, but not limited to, tasks like spinal diseases detection, patient rehabilitation results assessment or personal identification.

Table 4. Stride sub-phases, their average length and classification criteria.

Stride phase	Avg. end of interval	Classification criteria
Initial contact	2	Maximum value in stride cycle
Loading response	12	Cluster 1 following initial contact
Mid stance	31	Cluster 2 following Cluster 1
Terminal stance	50	Cluster 3
Pre-swing	62	Cluster 2 following Cluster 3
Swing	100	Cluster 1 following Cluster 2

Table 5. Stride cycle phase classification results.

Stride cycle	Length (samples)	Correctly classified samples	Accuracy
Stride cycle 1	113	98	86,7
Stride cycle 2	114	95	83,3
Stride cycle 1	227	193	85,0

(a) 70 degrees (b) 50 degrees

Fig. 14. Clustering results for two different data overlapping amounts (70 and 50 degrees).

(a) 30 degrees (b) 10 degrees

Fig. 15. Clustering results for two different data overlapping amounts (30 and 20 degrees).

8. Conclusion

In the paper a complete framework for quaternion clustering was presented, which consists of a clustering algorithm along with test data generation framework, quaternion clustering quality measures and the quaternion clustering number of the cluster selection algorithm.

The proposed quaternion clustering algorithm works fully in the quaternion domain. It employs random clusters generation, data assignment to clusters and quaternions averaging based on quaternions domain operations only (e.g., random quaternion generation and the quaternion distance measure). Similarly, clustering validity indices were adapted for use in the quaternion domain. The proposed visualization method allows us to observe that similar rotations described by quaternions are within the same cluster.

Three experiments were carried out using the provided methods: testing clustering against different numbers of clusters for randomly generated data, testing clustering against different degrees of data overlapping, testing clustering against recorded real gait kinematic data. All of those experiments were aimed at observing quaternion clustering and quality index properties. Results obtained from the first experiment are summarized in Table 1. One may conclude that the QDB, QDI and QCH indices employed all together in the voting algorithm described in Section 5.1 provide a reliable number of clusters for the tested data sets. It is worth noticing that the number of cluster selections was derived in an automated procedure.

It is worth highlighting that local criteria have been used, and it is to be noticed that selecting the number of clusters based on local minima and maxima of quality indices can provide a more adequate number of clusters. Figures 7 and 8 visualize clustering results for the data generated using the proposed method. Figure 9 compares the clustered data against that original randomly generated, and it looks like they are very much in-line. The clusters obtained from k-means clustering are similar to the originally generated clusters.

From the results of the additional experiment comparing clustering results obtained from quaternions k-means versus Euler angles baseline k-means, one can conclude that quaternion clustering seems to provide better clusters separability, which is an interesting observation pointing to further research on the subject.

The second test provides results helping to judge how valid the clustering criteria are with increasing data overlapping (data complexity). The proposed methodology performed well. Up to the data overlapping, an amount equal to 50 degrees of the real number of clusters was computed. When the input data overlapping amount was higher than 50 degrees, the estimation of the underlying number of clusters was very close to the actual one.

In the third experiment, all previously verified methods were tested on recorded real gait kinematic data. As presented in Figs. 16, 17 and 18, the proposed methodology allowed grouping similar rotations recorded from respective bones into a fixed number of clusters. Additionally, we conducted an experiment to automatically detect stride subphases based on the obtained gait data clustering. The phase classification task was performed with 85% accuracy, which is a solid basis for further research and practical applications of the proposed method.

The presented approach forms a quaternion clustering framework consisting of a quaternion k-means method, a test data generation method, a set of validity indices and an automated multi-index selection algorithm for an expected number of clusters. The performed set of experiments verified algorithm and index properties for artificial and real data sets. Specifically, we tested the approach for use with recorded real gait kinematic data. The presented framework can have multiple areas of applicability for motion capture and motion tracks analysis including, but not limited to, human motion similarity comparison and analysis of large datasets, motion-based person recognition techniques, detection of anomalies in motion track as a result of disorder or disease. Quaternion-based clustering can be also used as a part of more complex algorithms, for example, analysis of chaotic behaviour (Piórek, 2018).

Acknowledgment

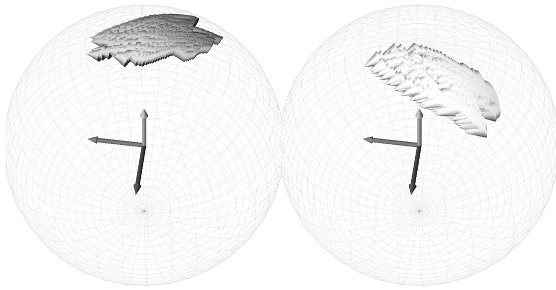
The gait data recordings were captured in the Center for Research and Development of the Polish-Japanese Academy of Information Technology (`www.bytom.pja.edu.pl`). The authors would like to express their deepest gratitude for access to these data sets.

References

Abhayasinghe, N. and Murray, I. (2014). Human gait phase recognition based on thigh movement computed using imus, *9th IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore*, pp. 1–4.

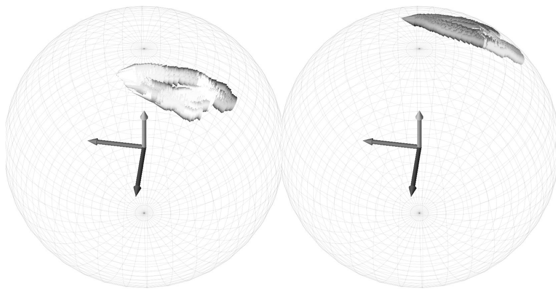
Austin, S.B., Melly, S.J., Sanchez, B.N., Patel, A., Buka, S. and Gortmaker, S.L. (2005). Clustering of fast-food restaurants around schools: A novel application of spatial statistics to the study of food environments, *American Journal of Public Health* **95**(9): 1575–1581.

Bandyopadhyay, S. and Maulik, U. (2002). Genetic clustering for automatic evolution of clusters and application to image classification, *Pattern Recognition* **35**(6): 1197–1208.



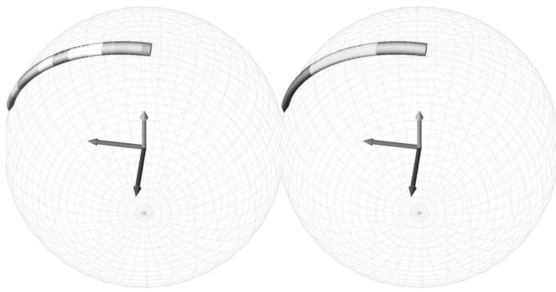
(a) left foot (b) right foot

Fig. 16. Visualization of clustering for live gait kinematic data (feet).



(a) left femur (b) right femur

Fig. 17. Visualization of clustering for live gait kinematic data (femurs).



(a) left tibia (b) right tibia

Fig. 18. Visualization of clustering for live gait kinematic data (tibias).

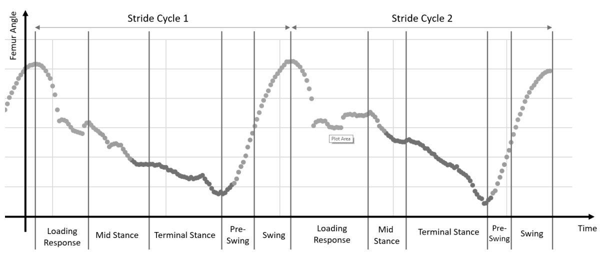


Fig. 19. Gait sub-phases analysis for live gait kinematic data (right femur).

Berkhin, P. (2006). A survey of clustering data mining techniques, *in* J. Kagan *et al.* (Eds), *Grouping Multidimensional Data*, Springer, Berlin/Heidelberg, pp. 25–71.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis, *Communications in Statistics* **3**(1): 1–27, DOI: 10.1080/03610927408827101.

Cantador, I. and Castells, P. (2006). Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering, *in* S. Staab and V. Svwtek (Eds), *Managing Knowledge in a Word of Networks*, Springer, Berlin/Heidelberg, pp. 334–349.

Chaturvedi, N.A., Sanyal, A.K. and McClamroch, N.H. (2011). Rigid-body attitude control, *IEEE Control Systems* **31**(3): 30–51.

Creusot, C., Pears, N. and Austin, J. (2010). 3D face landmark labelling, *Proceedings of the ACM Workshop on 3D Object Retrieval, Firenze, Italy*, pp. 27–32.

Davies, D.L. and Bouldin, D.W. (1979). A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**(2): 224–227, DOI: 10.1109/TPAMI.1979.4766909.

Dunn, J.C. (1974). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* **3**(3): 32–57.

Ell, T.A. and Sangwine, S.J. (2007). Hypercomplex Fourier transforms of color images, *IEEE Transactions on Image Processing* **16**(1): 22–35.

Gariel, M., Srivastava, A.N. and Feron, E. (2011). Trajectory clustering and an application to airspace monitoring, *IEEE Transactions on Intelligent Transportation Systems* **12**(4): 1511–1524.

Grubesic, T.H. (2006). On the application of fuzzy clustering for crime hot spot detection, *Journal of Quantitative Criminology* **22**(1): 77–105.

Han, J., Pei, J. and Kamber, M. (2011). *Data Mining: Concepts and Techniques*, Elsevier, Waltham, MA.

Himberg, J., Hyvärinen, A. and Esposito, F. (2004). Validating the independent components of neuroimaging time series via clustering and visualization, *Neuroimage* **22**(3): 1214–1222.

Huynh, D.Q. (2009). Metrics for 3D rotations: Comparison and analysis, *Journal of Mathematical Imaging and Vision* **35**(2): 155–164, DOI: 10.1007/s10851-009-0161-2.

Jabłoński, B. (2008a). Anisotropic filtering of multidimensional rotational trajectories as a generalization of 2D diffusion process, *Multidimensional Systems and Signal Processing* **19**(3): 379–399, DOI: 10.1007/s11045-008-0056-1.

Jabłoński, B. (2008b). *Filtration of Images and Spatial Trajectories Using Partial Differential Equations*, EXIT, Warsaw, (in Polish).

Jabłoński, B. (2011). Application of quaternion scale space approach for motion processing, *in* R.S. Choraś (Ed.), *Image Processing and Communications Challenges 3*, Springer, Berlin/Heidelberg, pp. 141–148, DOI: 10.1007/978-3-642-23154-4_16.

Jabłoński, B. (2012). Quaternion dynamic time warping, *IEEE Transactions on Signal Processing* **60**(3): 1174–1183.

Johnson, M. (2003). *Exploiting Quaternions to Support Expressive Interactive Character Motion*, PhD thesis, MIT, Cambridge, MA.

Koster, K. and Spann, M. (2000). MIR: An approach to robust clustering-application to range image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(5): 430–444.

Liao, T.W. (2005). Clustering of time series data—a survey, *Pattern Recognition* **38**(11): 1857–1874.

Loots, M.T., Bekker, A., Arashi, M. and Roux, J.J. (2013). On the real representation of quaternion random variables, *Statistics* **47**(6): 1224–1240, DOI: 10.1080/02331888.2012.695376.

Markley, F.L., Cheng, Y., Crassidis, J.L. and Oshman, Y. (2007). Averaging quaternions, *Journal of Guidance, Control, and Dynamics* **30**(4): 1193–1197.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(12): 1650–1654.

Pei, S.-C. and Cheng, C.-M. (1999). Color image processing by using binary quaternion-moment-preserving thresholding technique, *IEEE Transactions on Image Processing* **8**(5): 614–628.

Piórek, M. (2018). *Analysis of Chaotic Behavior in Non-linear Dynamical Systems*, Springer, Cham.

Reumerman, H.-J., Roggero, M. and Ruffini, M. (2005). The application-based clustering concept and requirements for intervehicle networks, *IEEE Communications Magazine* **43**(4): 108–113.

Risojević, V. and Babić, Z. (2013). Unsupervised learning of quaternion features for image classification, *11th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Serbia, Nis*, Vol. 1, pp. 345–348.

Shi, L. (2005). *Exploration in Quaternion Colour*, PhD thesis, Simon Fraser University, Burnaby.

Shi, L. and Funt, B. (2007). Quaternion color texture segmentation, *Computer Vision and Image Understanding* **107**(1): 88–96.

Wu, Z. and Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11): 1101–1113.

Zhou, F., De la Torre, F. and Hodgins, J.K. (2008). Aligned cluster analysis for temporal segmentation of human motion, *8th IEEE International Conference on Automatic Face & Gesture Recognition, FG'08, Amsterdam, The Netherlands*, pp. 1–7.

Michał Piórek received his MSc and PhD degrees in control engineering and robotics in 2013 and 2017, respectively, both from the Wrocław University of Science and Technology, Poland. He as a research assistant at the Department of Computer Engineering there. His research interests are in dynamical systems, chaos theory, processes described by quaternions, and nonlinear time series analysis.

Bartosz Jabłoński received his MSc degree in computer electronics and telecommunications in 2003 and his PhD degree in automation and robotics in 2007, both from the Wrocław University of Technology, Poland. In 2007, he became a researcher (since 2010 an assistant professor) in the Department of Control Systems and Mechatronics there. His research interests include different aspects of image processing, motion analysis, partial differential equations and quaternions.

Received: 4 August 2018
Revised: 23 January 2019
Re-revised: 12 June 2019
Accepted: 29 July 2019