Intelligent Data Processing and Its Applications

PhD Thesis

Anikó Szilvia Vágner

Supervisors:
Katalin Juhász
Márton Ispány

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola

Debrecen
2016

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács Informatikai Tudományok Doktori Iskola, Alkalmazott információ technológia és elméleti háttere programja keretében készítettem a Debreceni Egyetem Informatikai Tudományok doktori (PhD) fokozatának elnyerése céljából.

Debrecen, 2016. január 15.

Vágner Anikó Szilvia

Tanúsítom, hogy Vágner Anikó Szilvia doktorjelölt 2004 és 2012 között Dr. Nyakóné dr. Juhász Katalin, ezt követően 2012 és 2016 között a fent megnevezett Doktori Iskola Alkalmazott információ technológia és elméleti háttere programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult.

Az értekezés elfogadását javasolom.

Debrecen, 2016. január 15.

Dr. Ispány Márton

Intelligent Data Processing and Its Applications

Értekezés a doktori (PhD) fokozat megszerzése érdekében az Informatika tudományágban

Írta: Vágner Anikó Szilvia okleveles matematika-informatika tanár

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskolája (Alkalmazott információ technológia és elméleti háttere programja) keretében

Témavezetők:

Dr. Nyakóné dr. Juhász Katalin

Dr. Ispány Márton

A doktori szigorlati bizottság:

elnök:	Dr. Pethő Attila	.....
tagok:	Dr. Kiss Attila	.....
	Dr. Csernoch Mária	.....

A doktori szigorlat időpontja: 2015. november 15.

Az értekezés bírálói:

	Dr.	.....
	Dr.	.....

A bírálóbizottság:

elnök:	Dr.	.....
tagok:	Dr.	.....
	Dr.	.....
	Dr.	.....
	Dr.	.....

Az értekezés védésének időpontja: 20... . ...

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Nowadays the rapidly increasing performance of hardware and the efficient intelligent scientific algorithms enable us to store and process big data. This tendency will offer more opportunities to get more and more information from the large amount of data.

My thesis is only a precursor of this topic, because I did not have sufficient hardware and I had only a little data to be processed. However, all the topics of my thesis belong to the intelligent data processing.

In Chapter 2 I introduce a new clustering algorithm named GridOPTICS, whose goal is to accelerate the well-known OPTICS density clustering technique. The density-based clustering techniques are capable of recognizing arbitrary-shaped clusters in a point set. The DBSCAN results in only one cluster set, but the OPTICS generates a reachability plot from which a lot of cluster sets can be read as a result without having to execute the whole algorithm again. I experienced that it is very slow for large data sets, so I wanted to find a solution to accelerate it. I wanted to see that the speed of the GridOPTICS is better than OPTICS, so I executed both the algorithms on several point sets.

In Chapter 3 I introduce two new modules of the Cardiospy system of Labtech Ltd. On these two projects I worked together with István Juhász, László Farkas, Péter Tóth, and 4 students of the university, József Kuk, Ádám Balázs, Béla Vámosi, and Dávid Angyal. Béla Kincs, who was the executive of the Labtech Ltd., wanted the Cardiospy system to be improved. He and his team surveyed what the demand of the users are in this area and how their software could be better. The Labtech Ltd. and the University of Debrecen worked together in these two projects. In both cases the Labtech had early solutions for the algorithms, but they were inefficient and slow, the results could not be validated, or they gave insufficient results. Moreover, there were no visualization tools for either problems. The tasks of the team of the University of Debrecen were to give a quick algorithm and to create an interactive visualization interface for each problem.

The goal of the first module of Cardiospy is to cluster and visualize the long (up to 24-hours) recordings of ECG signals, because the manual evaluation of long recordings is a lengthy and tedious task. During this project I recognized that it is a very interesting topic to find out how the OPTICS can be accelerated with a grid clustering method independently, without any ECG signals.

The goal of the second module of Cardiospy is to calculate and visualize the steps of the blood pressure measurement and the values of blood pressure. The

recordings (which can contain a sequence of measurements) are collected by a microcontroller, but this module runs on a PC. With the help of the application the physicians can recognize the types of errors on the measurements and they can also find the noisy measurements.

In Chapter 4 I introduce how I applied an active learning method in a subject whose topic is database programming. I taught Oracle SQL and PL/SQL in the Advanced DBMS 1 subject and I saw that the students do not practice at home. The prerequirements of this subject are the Programming language and the Database systems courses, so they are not absolute beginners in the field. I wanted to force the students to try out the programming tools independently, but with the help of the teacher.

To support the active learning method, an application had to be developed. The application helps the teacher organize and monitor the tasks and their solutions of the students. Moreover, the application can verify the syntax of the solutions before the students upload them. If the syntax is wrong, the student cannot upload it. This feature makes the task of the teacher easier.

To demonstrate whether the active learning method is good or not, I gathered and examined the results of the students during the 3 years when I used this method.

2 Clustering algorithms

Cluster analysis is an important research field of data mining, namely an unsupervised data mining technique, which is applied in many other disciplines, such as pattern recognition, image processing, machine learning, bioinformatics, information retrieval, artificial intelligence, marketing, psychology, etc. Data clustering is a method of creating groups or clusters of objects in a way that objects in one cluster are very similar to each other and objects in different clusters are quite distinct. In data clustering, the classes are not predefined, clustering algorithms determine them. (Gan et al., 2007)

There are many more or less effective clustering algorithms, such as grid-based, hierarchical, fuzzy, center-based, search-based, graph-based, density-based, model-based, subspace clustering, etc. (Gan et al., 2007), (Han and Kamber, 2006)

2.1 Grid-based clustering techniques

The grid-based clustering creates a grid structure from the data points in the first step, in other words it partitions the large data points into a finite number of cells and calculates the cell density for each cell. The cells are predefined; the input data points do not impact its creation. In the next step, the algorithm operates on the grid structure to identify the clusters (Gan et al., 2007). The great advantage of grid-based clustering is its significant reduction of the computational complexity, especially for clustering very large data sets, which means, its processing time is fast, because similar data points will belong to the same cell and will be regarded as a single point. "This makes the algorithms independent of the number of data points in the original data set." (Gan et al., 2007). Well-known grid-based clustering techniques are the OptiGrid (Hinneburg and Keim, 1999), CLIQUE (Agrawal et al., 1998) and the WaveCluster (Seikholeslami et al., 2000). Han and Kamber (2006) and Gan et al., (2007) gave a comprehensive summary of these techniques.

2.1.1 OptiGrid - Optimal Grid-Clustering

The algorithm works recursively. In each step, if it is possible, it partitions the actual data set into subsets by using maximum q cutting planes. The cutting planes are orthogonal to at least one projection and they are chosen to have a minimal point density. The recursion of a subset stops if there are no more good cutting planes.

The algorithm is efficient for large, high-dimensional data sets with noise. However, it may be slow, because it uses a recursive method. (Hinneburg and Keim, 1999), (Gan et al., 2007)

2.1.2 CLIQUE - CLustering InQUEst

This method can be considered as a combination of density-based and grid-based clustering methods, because it partitions each dimension like a grid structure and determines whether a cell is dense based on the number of points it contains. Firstly, CLIQUE partitions the dimensional data space into non-overlapping rectangular units and identifies the dense units. A unit is dense if the fraction of total data points contained in it exceeds a parameter. This is done for each dimension. Then it determines candidate search spaces which consist of the subspaces representing the dense units and in which dense units of higher dimensionality may exist.

In the next step, for each cluster, CLIQUE determines the maximal region that covers the cluster of connected dense units and finally it determines a minimal cover for each cluster.

The algorithm is insensitive to the order of input objects. But, the accuracy of the clustering results may be degraded because of the simplicity of the method. Furthermore, using this algorithm it is difficult to find clusters of different density within different dimensional subspaces. (Agrawal et al., 1998), (Han and Kamber, 2006)

2.1.3 WaveCluster

WaveCluster is a multiresolution, grid-based, and density-based clustering algorithm. Firstly, it builds a multidimensional grid structure on the data space. The grid structure consists of nonoverlapping hyperrectangles. Then the algorithm summarizes the information of a group of points that map into a grid cell.

In the next step, it uses a wavelet transformation on the grid structure for each dimension of the space after each other. The wavelet transform is a signal processing technique which decomposes a signal into different frequency subbands. The wavelet transformation ignores some information, but it preserves the relative distance between two objects. As a result, the algorithm gives labels to each grid cell. Based on these labels, the algorithm creates the clusters.

This clustering method efficiently handles large data sets, it recognizes the arbitrary-shaped clusters, it handles noise, it is insensitive to the order of input points, and it can be applied for multidimensional data sets. But it is efficient only for low-dimensional data sets. (Seikholeslami et al., 2000), (Han and Kamber, 2006), (Gan et al., 2007)

2.2 Density-based clustering methods

The density-based clustering approach is capable of finding arbitrarily shaped clusters. The clusters are dense regions, which are separated by sparse regions. These algorithms can handle noise very efficiently. "The number of clusters is not required as a parameter, since density-based clustering algorithms can automatically detect the clusters" (Gan et al., 2007), and in this way they determine the number of the clusters as well. There is a disadvantage of the most density-based techniques that it is hard to choose parameter values in order that the algorithm can give an appropriate result. (Gan et al., 2007), (Han and Kamber, 2006)

It is not easy to define when it can be said that the result of a clustering algorithm is appropriate. There can be two extreme cases, when all points are noise and when all points belong to the same cluster. But they cannot be accepted as appropriate results (except for special point set), because the users expect what they perceive on the point set. The main problem is that the users have subjective viewpoints, so on the same point set one user will see two clusters, and the other one will see four clusters. All in all the result of a clustering method is appropriate if the user can perceive clusters in the point set without any algorithms, the algorithm recognizes similar clusters in it.

Gan et al. (2007) and Han and Kamber (2006) reviewed the well-known density-based clustering algorithms, which are the DENCLUE (Hinneburg and Keim, 1998), (Hinneburg and Gabriel, 2007), DBSCAN (Ester et al., 1996), and the OPTICS (Ankerst et al., 1999).

2.2.1 DENCLUE - DENsity-based CLUstEring

DENCLUE method is based on a set of density distribution functions. It uses the following concepts:

– An *influence function* of a point Y at a point X is a mathematical function which is used to formally model the impact of the data point Y within its neighborhood. The influence function can be an arbitrary function that can

be determined by the distance between two objects in a neighborhood. The distance function should be reflexive and symmetric.
- The *density function* at a point X is defined as the sum of influence functions of all data points at a point X. The overall density of the data space is analytically modeled in this way.
- The *density attractor* is the local maxima of the overall density function. For a continuous and differentiable influence function, a hill-climbing algorithm can be used to determine the density attractor of a set of data points. The clusters can be determined if the density attractors are identified.
- The neighborhood of a density attractor with some other conditions is called *density-attracted* points. In general, points that are density attracted to a density attractor may form a cluster.
- The *center-defined cluster* for a density attractor X^* is a subset of all points that are density-attracted by X^*, and where the density function at X^* is no less than a threshold, ξ. The points that are density-attracted by X^*, but for which the density function value is less than ξ are considered noise.
- The *arbitrary-shape cluster* is subset of each point that is density-attracted to a density attractor at which the density function value is no less than a threshold, ξ, and there exists a path from each density-attractor to another, where the density function value for each point along the path is no less than ξ.

The method has a great mathematical foundation. It handles data sets with large amount of noise well.

The method requires careful selection of the density parameter and noise threshold, because the values of these parameters may significantly impact the quality of the clustering results. (Hinneburg and Keim, 1998), (Han and Kamber, 2006)

2.2.2 DBSCAN - Density-Based Spatial Clustering of Applications with Noise

DBSCAN (Ester et al., 1996) grows regions with sufficiently high density into clusters. It defines a cluster as a maximal set of density-connected points. The key idea of the method is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighborhood has to exceed some threshold. The method works with any distances.

It uses the following definitions:

- The ϵ-*neighborhood* of an object is the neighborhood within a radius ϵ of the object.
- A *core object* is the objects whose ϵ-neighborhood contains at least a minimum number, $MinPts$, of objects.
- An object P is *directly density-reachable* from object Q if P is within the ϵ-neighborhood of Q and P is a core object.
- An object P is *density-reachable* from object Q with respect to ϵ and $MinPts$ in a set of objects, D, if there is a chain of objects $P_1, \ldots, P_n$, where $P_1 = Q$ and $P_n = P$ such that P_{i+1} is directly density-reachable from P_i with respect to ϵ and $MinPts$, for $i = 1 \ldots n$, $P_i \in D$.
- An object P is *density-connected* to object Q with respect to ϵ and $MinPts$ in a set of objects, D, if there is an object $O \in D$ where both P and Q are density-reachable from O with respect to ϵ and $MinPts$.

Density-reachability is the transitive closure of direct density-reachability (the transitive closure of a binary relation R on a set X is the transitive relation R^+ on set X such that R^+ contains R and R^+ is minimal (Lidl and Pilz, 1998)). The density-reachability relationship is asymmetric. Only core objects are mutually density-reachable. Density-connectivity is a symmetric relation.

DBSCAN defines the density-based cluster as a set of density-connected objects that is maximal with respect to density-reachability. Every object that is not contained in any cluster is considered as noise.

A cluster C with respect to ϵ and $MinPts$ contains at least $MinPts$ points.

The algorithm has two main steps, which are repeated while there are unprocessed points. Firstly, it chooses an arbitrary point from the database satisfying the core point condition as a seed. Secondly, it retrieves all points that are density-reachable from the seed obtaining the cluster containing the seed. If there are no more points in the cluster, the algorithm repeats the first step, if there are unprocessed points.

Algorithm 1 shows the pseudo-code of the DBSCAN algorithm.

DBSCAN has a problem when clusters have widely varying densities. Moreover, it can be expensive, because determining the nearest neighbors needs the calculation of all point pairs. Finally, it is sensitive for input parameters. (Ester et al., 1996), (Han and Kamber, 2006), (Tan et al., 2005)

Algorithm 1 The pseudo-code of DBSCAN

```
ClusterId = 0;
while Element Number of Unprocessed Elements != 0 do
    C is an Element From the Unprocessed Elements;
    account C processed;
    if C is core-object then
        ClusterId of C = ClusterId;
        add Neighbors of C to Neighbor Elements;
        ClusterId of Neighbors of C = ClusterId;
        while Element Number of Neighbor Elements != 0 do
            S is an Element from Neighbor Elements;
            account S processed;
            take out S from Neighbor Elements;
            if S is core-object then
                add Unprocessed Neighbors of S to Neighbor Elements;
                ClusterId of Neighbors of S = ClusterId;
        increase ClusterId;
    else
        ClusterID of C = Noise;
```

2.2.3 OPTICS

Clustering algorithms are sensitive to input parameters, in other words they have a significant influence on the results of clustering. It is not easy to find the parameters which ensure satisfying results. "The OPTICS algorithm creates an augmented ordering of the database representing its density-based clustering structure. This cluster-ordering contains information which is equivalent to the density-based clustering corresponding to a broad range of parameter settings." (Ankerst et al., 1999). The cluster ordering can be used to extract basic clustering information (such as cluster centers or arbitrary-shaped clusters) as well as provide the intrinsic clustering structure.

In the case of DBSCAN, for a constant $MinPts$ value, density-based clusters with respect to a lower value for ϵ are completely contained in density-connected sets obtained with respect to a higher value for ϵ.

Therefore, in order to produce an ordering of density-based clusters, DBSCAN algorithm was extended to process a set of distance parameter values at the same time. To construct the different clusterings simultaneously, the objects have to be processed in a specific order. This order selects an object that is density-reachable with respect to the lowest ϵ value so that clusters with lower ϵ will be finished first. (Ankerst et al., 1999), (Han and Kamber, 2006)

OPTICS need to store two values for each object, they are the core-distance and the reachability-distance. The *core-distance* of the point C is the smallest ϵ' ($\epsilon' \leq \epsilon$) of which it is true that the cardinality of the ϵ'-neighborhood of the point C is equal or greater than $MinPts$; if this ϵ' does not exist, it is undefined. The *reachability-distance* of point P with regard to point C is undefined if the C is not core-object, otherwise the greater value from the core-distance of point C and the distance of point P and point C. (Ankerst et al., 1999)

The OPTICS algorithm generates a structure in which the sequence of the input points is important, and it assigns a corresponding reachability-distance for each point. This structure can be displayed by a 2-D plot, whose name is reachability plot. Valleys in the reachability plot indicate clusters: points having a small reachability value are closer and thus more similar to their predecessor points than points having high reachability value. (Brecheisen et al., 2006)

Algorithm 2 shows the pseudo-code of the OPTICS algorithm.

Algorithm 2 The pseudo-code of OPTICS

```
1:  Calculate Core Distances;
2:  while Element Number of Unprocessed Elements != 0 do
3:      C is an Element From the Unprocessed Elements;
4:      account C processed;
5:      if C is core-object then
6:          add Neighbors of C to Neighbor Elements;
7:          while Element Number of Neighbor Elements != 0 do
8:              calculate Reachability Distances for every Element of Neighbor
                    Elements with regard to each Processed Element;
9:              S is the Element from Neighbor Elements which has the Smallest
                    Reachability Distance;
10:             account S processed;
11:             take out S from Neighbor Elements;
12:             if S is core-object then
13:                 add Neighbors of S to Neighbor Elements;
```

If you want to determine the clusters of this structure, you can use the algorithm of Ankerst et al. (1999). They generate a hierarchical clustering structure from the reachability plot. They give some definitions which help in the identification of the clusters.

Valleys in a reachability plot indicate clusters. A valley begins with steep downward points and ends with steep upward points. A new parameter ξ is used in order that the degree of the steepness can be defined. A point is a ξ-*steep*

upward point if it is ξ% lower than its successor. Point P is a *ξ-steep downward* point if its successor is ξ% lower than P.

More precisely, a valley begins with a steep downward area and ends with steep upward area. An interval $I = [s, e]$ is *ξ-steep upward area*, if s is ξ-steep upward point, e is ξ-steep upward point, each point between s and e is at least as high as its predecessor, it does not contain more than $MinPts$ consecutive points that are not ξ-steep upward and I is maximal. A *ξ-steep downward area* is defined analogously.

With the help of the previous definitions, the cluster can be defined. In the definition, $r(x)$ is the reachability distance of x. An interval $C = [s, e]$ is a ξ-cluster if $\exists D = [s_D, e_D], U = [s_U, e_U]$ where

1. D is ξ-steep downward area and $s \in D$,
2. U is ξ-steep upward area and $e \in U$,
3. $e - s \geq MinPts$,
4. $\forall x, s_D < x < e_U : (r(x) \leq min(r(s_D), r(e_U)) \times (1–\xi/100))$,
5. $(s, e) =$
$$\begin{cases} (max\{x \in D : r(x) > r(e_U + 1)\}, e_U) & if\ r(s_D) \times (1–\xi/100) \geq r(e_U + 1) \\ (s_D, min\{x \in U : r(x) < r(s_D\}) & if\ r(e_U + 1) \times (1–\xi/100) \geq r(s_D) \\ (s_D, e_U)\ otherwise \end{cases}$$

Algorithm 3 shows the pseudo-code of the automatic cluster recognizer for a reachability plot introduced by Ankerst et al. (1999).

Patwary et al. (2013) provided a simpler algorithm to find the clusters based on the reachability plot. They use a new φ parameter ($0 \leq \varphi \leq \epsilon$). Their idea is that two points X and Y belong to the same cluster if X is directly density reachable from Y which is a core point. The first point of a cluster has greater reachability distance than φ and it is a core point (its core distance is less than φ). If the two conditions are satisfied, the algorithm begins a new cluster and keeps adding the following points, Y as long as Y is directly density reachable from any of the previously added core points in the same cluster, that is, reachability distance of Y is not greater than φ. Any points not part of a cluster are declared as NOISE points. Algorithm 4 shows the pseudo-code of their algorithm.

2.3 Combinations of the grid-based and density-based techniques

Combination of the grid-based and the density-based technique is common. Parikh and Varma (2014) gave a short survey of this topic. They presented short

descriptions of some grid-based algorithms namely the new shifting clustering algorithm, the grid-based DBSCAN algorithm, the GDILC algorithm, the general grid-clustering approach, and the OPT-GRID(S).

Algorithm 3 The pseudo-code of the automatic cluster recognizer for a reachability plot

```
1: SetOfSteepDownAreas = EmptySet;
2: SetOfClusters = EmptySet;
3: index = 0;
4: mib = 0;
5: while index < n do
6:     mib = max(mib, r(index));
7:     if start of a steep down area D at index then
8:         update mib-values and filter SetOfSteepDownAreas;
9:         set D.mib = 0;
10:        add D to the SetOfSteepDownAreas;
11:        index = end of D + 1;
12:        mib = r(index);
13:    else
14:        if start of steep up area U at index then
15:            update mib-values and filter SetOfSteepDownAreas;
16:            index = end of U + 1;
17:            mib = r(index);
18:    for each D in SetOfSteepDownAreas do do
19:        if combination of D and U is valid and
20:            satisfies cluster conditions 1, 2, 3 then
21:            compute [s, e] add cluster to SetOfClusters;
22:        else
23:            index = index + 1;
24: return SetOfClusters;
```

Similarly, Mann and Kaur (2013) collected some DBSCAN variant algorithms, namely GMDBSCAN and GDCLU combined the two techniques. The grid-based DBSCAN algorithm (Darong and Peng, 2012) is similar to GridOPTICS algorithm; however, they improved the DBSCAN algorithm.

G-DBSCAN (Ma et al., 2014) uses a grid method for the first time, and it removes noise in order to reduce the points to be processed. Its goal was to reduce memory usage and improve efficiency of the algorithm. They did not give exact information on how they assign input points to the grid structure, moreover, they analyzed the efficiency of their algorithm on data sets which have only about a few hundred points. Zhao et al. (2011) proposed an enhanced grid-density based approach for clustering high dimensional data, which was accurate and fast, which they showed

in the experimental evaluation, where they executed their AGRID+ algorithm for more synthetic data sets. Ma et al. (2003) presented the CURD algorithm, which uses references and density, and which has nearly linear time complexity. Achtert et al. (2006) introduced the DeLiClu algorithm, which avoids the non-intuitive ϵ parameter of the OPTICS and the density estimator of the single-link algorithm.

Algorithm 4 The pseudo-code of the cluster recognizer of Patwary et al. (2013)

```
1: ClusterId=0;
2: for x = 1 .. n do
3:     if reachabilitydistance(x) > φ then
4:         if coredistance(x) ≤ φ then
5:             ClusterId = ClusterId + 1;
6:             ClusterId(x) = ClusterId;
7:         elseClusterId(x) = NOISE;
8:     elseClusterId(x) = ClusterId;
```

Some researchers changed the OPTICS algorithm in order to improve it in a way. Schneider and Vlachos (2013) introduced the fast density-based clustering technique based on random projections (FOPTICS), whose goal was to speed up the computation of the OPTICS algorithm. Alzaalan et al. (2012) enhanced the concept of core-distance of the OPTICS in order to make the algorithm less sensitive to data with variant density but they did not improve the performance. Patwary et al. (2013) introduced the scalable parallel OPTICS algorithm (POPTICS), which they tried out on a 40-core shared-memory machine. Brecheisen et al. (2006) confined the OPTICS algorithm to ϵ-range queries on simple distance functions and carried out complex distance computations only at a stage of the algorithm where they were compulsory to compute the correct clustering result. Breunig et al. (2000) combined compression, namely the BIRCH algorithm, with the OPTICS algorithm to yield performance speed-up-factors. Brecheisen et al. (2006) combined the multi-step query processing with density-based clustering algorithms, namely with the DBSCAN and the OPTICS in order to accelerate them by more than one order of magnitude.

Yue et al. (2007) presented a new algorithm named OGTICS, which modifies and improves the OPTICS with grid technology and has linear complexity, thus it is much faster than the OPTICS. The algorithm divides the data set into number of grids and assigns all data into these grids, then it partitions all grids to a few groups. In the next steps, it computes the center of each grid and orders all grids to a queue in the x axis. In the last two steps, it generates a statistical histogram with the number of data across all grids and determines the optimal number of clusters and partitions.

The GridOPTICS algorithm differs from this algorithm in creating of the grid structure and in the processing of the grid structure. Namely, my algorithm builds a simpler grid structure, and I use the OPTICS algorithm with only a few changes in the processing, whereas they used ordering the grids by x axis.

2.4 The GridOPTICS algorithm

I introduced a new clustering algorithm named GridOPTICS (Vágner, in press) which is a combination of a grid clustering and the OPTICS algorithm. GridOPTICS builds a grid structure to reduce the number of data points, then it applies the OPTICS clustering algorithm on the grid structure. In order to get the clusters, the algorithm uses the reachability plot of the grid structure, then it determines to which cluster the original input points belong. The new algorithm has the advantage that it has faster processing time than OPTICS, whereas it also keeps the advantages of the OPTICS.

2.4.1 Concepts used in GridOPTICS

GridOPTICS uses several concepts of OPTICS with the same meaning. The *neighbors* of the point C are the points which are in the ϵ-neighborhood of the C. Point C is a *core-object* if the cardinality of the ϵ-neighborhood of the point C is equal or greater than $MinPts$. The *core-distance* of the point C is the smallest ϵ' ($\epsilon' \leq \epsilon$) of which it is true that the cardinality of the ϵ' neighborhood of the point C is equal or greater than $MinPts$; if this ϵ' does not exist, it is undefined. The *reachability-distance* of point P with regard to point C is undefined if the C is not core-object, otherwise the greater value from the core-distance of point C and the distance of point P and point C.

2.4.2 The algorithm of the GridOPTICS

The main idea of the GridOPTICS algorithm is to reduce the number of input points with a grid technique and then to execute the OPTICS algorithm on the grid structure. Based on the reachability plot, the clusters of the grid structure can be determined. In the end, the input points can be assigned to the clusters. It is supposed that the points are in the Euclidean space, so the Euclidean distance is used, however, other distances can also be used. The GridOPTICS algorithm has 3 parameters, namely ϵ, $MinPts$, which play similar role as in the OPTICS, and τ, which defines the distance in the grid structure.

The algorithm has 4 main steps, they are the following:

1. Step: Constructing the grid structure

The grid structure is very simple, that is there are grid lines which are parallel and their distance is τ in a dimension, moreover, they are orthogonal to each other if they are not in the same dimension. In this way, they cross each other in grid points. The distance of two neighbor grid points is τ.

In an n-dimensional space, an input point $(p_1, p_2, \dots p_n)$ is assigned to a grid point $(g_1, g_2, \dots g_n)$ ($g_i = k_i * \tau$, k_i is an element of integers, $i = 1, \dots, n$) in the following way: $g_i - \tau/2 \leq p_i < g_i + \tau/2$, $(i = 1, \dots, n)$. The algorithm counts how many input points belong to each grid point. It only stores the grid points to which at least one input point has been assigned.

Figure 1 shows a simple example how the algorithm assigns the input points to the grid in two dimensions. There are the input points and the grid lines on the left side, whereas on the right side, there are the grid points and the number which shows how many input points belong to each grid point.

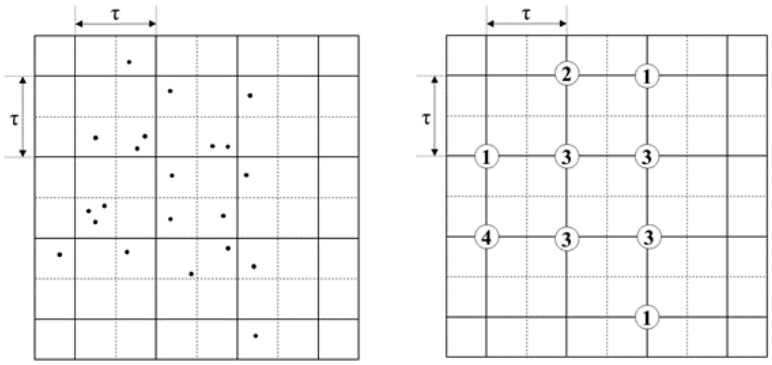

Figure 1: Input points and grid points

In this way, each input point is transformed into a grid point, which can hurt accuracy in a minimal way. If the grid was moved with τ' $(-\tau < \tau' < \tau)$ in a dimension, an input point would be likely to be transformed into another grid point but the inaccuracy problem would be the same. You will see that it could influence the results only to the slightest degree.

2. Step: Applying the OPTICS algorithm to the grid structure

The OPTICS searches the points in the ϵ-neighborhood of a point more times. To perform this task it should examine all input points. Because of the grid, this task

is simpler than in the OPTICS, since the neighbors of a grid point are also in the grid structure. We know that the distance of two neighbor grid points is τ, and we want to find points in ϵ-neighborhood of a point. Figure 2 shows the neighbor grid points of the C. The serial numbers of the grid points show the order in which the algorithm should process them when it calculates the core-distance of point C.

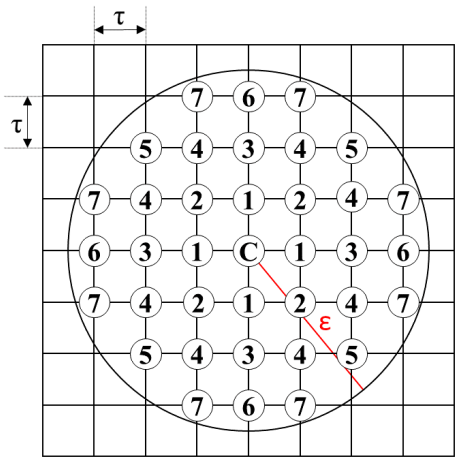

Figure 2: Point C and its neighbors

Zhao et al. (2011) gave a comprehensive discussion about the neighborhood on a high-dimensional grid structure.

In the second step, the algorithm calculates the core-distance of each stored grid point (point C) firstly. The OPTICS defines the core-distance of point C as the smallest ϵ' ($\epsilon' \leq \epsilon$) of which it is true that the cardinality of the ϵ' neighborhood of the point C is equal or greater than $MinPts$; if this ϵ' does not exists, it is undefined. The GridOPTICS algorithm calculates it in the next way:

1. if the number of the points assigned to the C is not less than $MinPts$, the core-distance will be 0;
2. if the distance of the C and the points marked by 1 on Figure 2 (which is τ) is not more than ϵ, the number of input points assigned to the C , and the grid points marked by 1 is not less than $MinPts$, the core-distance will be the distance of the C and the points marked by 1 (in this case this is τ);
3. if the distance of C and the points marked by I ($I = 2, 3, \dots$) on Figure 2 is not more than ϵ, the number of input points assigned to the C, and the grid points marked by $1, 2, \dots I$ is not less than $MinPts$, the core-distance will be the distance of the C and points marked by I;

4. otherwise the core-distance is undefined.

The other part of the second step is almost the same as the steps of the OPTICS. The algorithm chooses a grid point from the unprocessed grid points, and accounts it processed, then if it is a core-object, the algorithm continues the processing with the neighbor grid points, otherwise the algorithm repeats this step until there are unprocessed grid points.

Algorithm 5 The pseudo-code of the second step - Applying the OPTICS to the grid structure

```
1: Calculate Core Distances;
2: while Element Number of Unprocessed Grid Elements != 0 do
3:     C is an Element From the Unprocessed Grid Elements;
4:     account C processed;
5:     if C is core-object then
6:         add Neighbors of C to Neighbor Grid Elements;
7:         while Element Number of Neighbor Grid Elements != 0 do
8:             calculate Reachability Distances for every Element of Neighbor Grid
                   Elements with regard to each Processed Element;
9:             S is the Element from Neighbor Grid Elements which has the
                   Smallest Reachability Distance;
10:            account S processed;
11:            take out S from Neighbor Grid Elements;
12:            if S is core-object then
13:                add Neighbors of S to Neighbor Grid Elements;
```

In the processing of the neighbor grid points, the algorithm searches the neighbor grid points, and puts them into a neighbor collection. Then it chooses the point of the collection which has the smallest reachability-distance, accounts it processed, takes out from the neighbor collection, and if the point is core-object, the algorithm adds its neighbor grid points to the neighbor collection. Until the neighbor collection is not empty, the algorithm continues the processing of the next element of the neighbor collection.

Algorithm 5 shows the pseudo-code of the second main step.

As a result, there is a structure in which there is a given sequence of grid points with their corresponding reachability-distances.

3. Step: Determining clusters of the grid points

In this step, the algorithm assigns a cluster number to each cluster. I do not find automatically all clusters as Ankerst et al. (1999), instead I follow the method of Brecheisen et al. (2006), moreover, my algorithm is similar to the algorithm of

Patwary et al. (2013) but it is not the same. I also used the results of Sander et al. (2003), who automatically determined the significant clusters in the reachability plot with the help of dendrograms.

Algorithm 6 The pseudo-code of the third step - Determining clusters of the grid points

```
1: ClusterNumber = 0;
2: ClusterElementNumber = 1;
3: ClusterGridElementNumber = Number of Input Points of Processed Grid
   Elements [0];
4: Cluster Number of Processed Grid Elements [0] = clusterNumber;
5: for i = 1 .. Element Number of Processed Grid Elements do
6:     if Reachability Distance of Processed Grid Elements[i] ≥ φ then
7:         if (ClusterElementNumber < MinPts then
8:             for j = 0 .. clusterElementNumber  do
9:                 Processed Grid Elements [i - 1 – j] is NOISE;
10:        else
11:            ClusterNumber++;
12:        Cluster Number of Processed Grid Elements [i] = ClusterNumber;
13:        ClusterElementNumber = 1;
14:        ClusterGridElementNumber = Number of Input Points of Processed
               Grid Elements [i];
15:    else
16:        Cluster Number of Processed Grid Elements [i] = ClusterNumber;
17:        ClusterElementNumber += Number of Input Points of Processed Grid
               Elements [i];
18:        ClusterGridElementNumber++;
19: if ClusterElementNumber < MinPts then
20:    for j = 0 .. ClusterElementNumber do
21:        Processed Grid Elements [Processed Grid Elements Number - 1- j] is
               NOISE;
```

The goal is to find the clusters in which the reachability distance is less than φ ($0 \leq \varphi \leq$maximum of the reachability distances, or φ is undefined). You could also find all clusters if you applied all φ values but I will show results only for a few φ values. However, if you need all clusters, you can apply the algorithm of Ankerst et al. (1999) on the processed grid points of the GridOPTICS.

My algorithm processes the sequence of the grid points, and it says that if the reachability distance of a point is bigger than φ, there is a new cluster. However, if a cluster has fewer input points in the grid points than $MinPts$, it is noise, so it should examine how many points the cluster under processing has.

Algorithm 6 shows the pseudo-code of the third step.

4. Step: Assigning the input points to the clusters

In the last step, the GridOPTICS algorithm determines to which cluster each input point belongs. It looks through the input points, searches the grid point which was assigned to it, and reads its cluster number.

If you want to find all clusters in the reachability plot, you should execute the last two steps for all possible φ values.

2.4.3 Implementation

I realized the algorithm in the C# language in the Visual Studio 2010 Express Edition. I executed my program on two-dimensional input points.

I created the grid structure in a way that I shifted the input points in order that the minimum coordinates could be 0, and I divided the coordinates by τ. Consequently, the indexes of the grid structure started with 0 and its coordinates are small non-negative integer values, which makes the calculation faster.

In the core-distance calculation, I could use that the neighbor grid points of $C(cx, cy)$ marked by 1 on Figure 2 are $(cx - 1, cy)$, $(cx + 1, cy)$, $(cx, cy - 1)$, $(cx, cy + 1)$. Similarly, it is very easy to find the coordinates of the other neighbor grid points. To make the search of the neighbors easy, I used a List collection to store the relative coordinates of the possible ϵ-neighbors in the appropriate order shown in Figure 2, and I stored the grid structure in a Dictionary collection, which supported the search by coordinates.

The neighbor and the processed grid points were stored in List collections because the algorithm did not need to search the grid points by coordinates any more, but it needed the order of the processed grid points to produce the reachability plot.

2.4.4 A basic example

You can examine the steps of the GridOPTICS algorithm with a synthetic point set which has 20 points. Table 1 shows the input points named PointSetFirstTry20. The parameters of the algorithms are $\epsilon = 50$, $MinPts = 3$, $\tau = 4$.

In the first step, the GridOPTICS algorithm creates the grid structure, which is shown in Table 2. There are the two coordinates of the grid structure in the first

two columns, whereas the third column shows how many input points belong to the grid point.

Original x coordinate	Original y coordinate	Cluster number if $\varphi = 8$	Cluster number if $\varphi = 22$
54	7	1	1
2	5	noise	noise
30	3	noise	1
52	3	1	1
52	5	1	1
51	5	1	1
28	19	2	1
27	18	2	1
27	19	2	1
27	18	2	1
23	16	2	1
92	33	3	2
90	34	3	2
96	35	3	2
97	34	3	2
91	36	3	2
96	33	3	2
96	31	3	2
92	34	3	2
95	35	3	2

Table 1: The FirstTry20 point set

In the second step, firstly the algorithm calculates the core distances, which is shown in the fourth column. The core distances are calculated for the input points, namely the distances of the grid points are multiplied with the value of τ. Then the algorithm orders the points with the help of their reachability distances. The fifth column of Table 2 shows the reachability distances; moreover, the sequence of the grid points in Table 2 corresponds to the result of the algorithm. This means that the sequence of the grid points and their reachability distances constitute the reachability plot.

In the third step the algorithm assigns a cluster number to each clusters of grid points. The sixth and seventh columns of Table 2 show the cluster numbers when $\varphi = 8$ and $\varphi = 22$.

In the last step, the algorithm determines to which cluster each input point belongs. The third and fourth columns of Table 1 show the cluster numbers of input points. The order of the input points is the same as the original order, the reachability plot cannot be read from there.

x	y	Card. of input points	Core Distance	Reach. Distance	Cluster Number if $\varphi = 8$	Cluster Number if $\varphi = 22$
13	1	1	5,656854	3,402823E+38	1	1
12	0	3	0	5,656854	1	1
7	0	1	16,49242	20	noise	1
5	3	1	5,656854	14,4222	2	1
6	4	4	0	5,656854	2	1
0	0	1	28	28	noise	noise
22	8	4	0	45,60702	3	2
23	8	1	4	4	3	2
24	8	3	0	4	3	2
24	7	1	4	4	3	2

Table 2: The grid points with the cardinality of the input points, the reachability distances, the core distances, and the cluster numbers generated from the FirstTry20 point set ($\epsilon = 50$, $MinPts = 3$, $\tau = 4$)

2.4.5 Experimental results

Firstly, I used some home-generated synthetic point sets as input point sets of my GridOPTICS algorithm. I created the points by hands or I generated random points in several plan figures. The coordinates of the input points are integer values. The synthetic point sets have noisy points in order to show how the GridOPTICS finds them. Moreover, they have more clusters and these clusters have various densities in order that there can be more valleys of the reachability plot. The goal of the valleys is that we could easier make comparison between the reachability plot resulted by OPTICS and GridOPTICS. The program is executed on a PC which had 2GB RAM and 2.01 GHz AMD Athlon CPU.

I make comparison of execution time and results of the OPTICS and the GridOPTICS algorithm. Brecheisen et al. (2006) state that "ϵ has to be very high in order to create a reachability plot without loss of information and $MinPts$ is typically only a small fraction of cardinality of input data, e.g., $MinPts = 5$ is a suitable value even for large databases". In the experience, I used very large

ϵ values in order to easily compare the reachability plots; moreover, I also used higher $MinPts$ values.

The following abbreviations are used in the tables: OT is the execution time of the OPTICS, GOT is the execution time of the GridOPTICS, NGP is the number of the grid points, and MP is $MinPts$. The measure of the execution time is millisecond.

ϵ	MP	OT	τ	1	5	10	20	30	40
			NGP	391	225	100	55	39	18
500	5	6270ms	GOT	9392ms	2122ms	166ms	84ms	34ms	33ms
500	10	5583ms	GOT	9598ms	1512ms	211ms	76ms	11ms	6ms
500	20	6690ms	GOT	9598ms	1367ms	202ms	57ms	9ms	6ms

Table 3: The execution time of the algorithms on the PointSet500 data set

First I executed both algorithms on a synthetic point set with 407 input points called PointSet500. The x coordinates range between 84 and 573, whereas y coordinates are between 410 and 815. Table 3 shows the execution time of the OPTICS and the GridOPTICS and the number of grid points of the GridOPTICS.

If $\tau = 1$ the grid structure is almost the same as the input points, so the GridOPTICS will have almost the same result as the OPTICS but the execution will be longer because the GridOPTICS builds the grid structure first. If τ is bigger, the grid structure will have fewer grid points, so it will be faster. You can read from Table 3, if $\tau \geq 10$, the execution time will be less with one or two orders of magnitude or more.

Figure 3 shows the results executed on the PointSet500 data set with $\epsilon = 500$, $MinPts = 5$. Subfigure A shows the reachability plot produced by the OPTICS, Subfigure B is the clustered points resulted the OPTICS where $\varphi = 44$. The C, D, E, F, G, and H parts of the figure show the reachability plots produced by the GridOPTICS, where $\tau = 1, 5, 10, 20, 30$, and 40. The red line in reachability plots of each subfigures shows the value of the φ, which is 44 in these cases.

You can see little green points on the reachability plots of the GridOPTICS in Figure 3, which show how many input points belong to a grid point. The reachability plot of the GridOPTICS executed with $\tau = 1$ parameter are similar to that of the OPTICS, the algorithm can produce almost the same results as the OPTICS. The other reachability plot shows us that if the τ is higher, the plot is plainer, there are not so many cuts in the valleys, moreover, more input points belong to a grid point. Consequently the reachability distances in the valleys are growing with the τ.

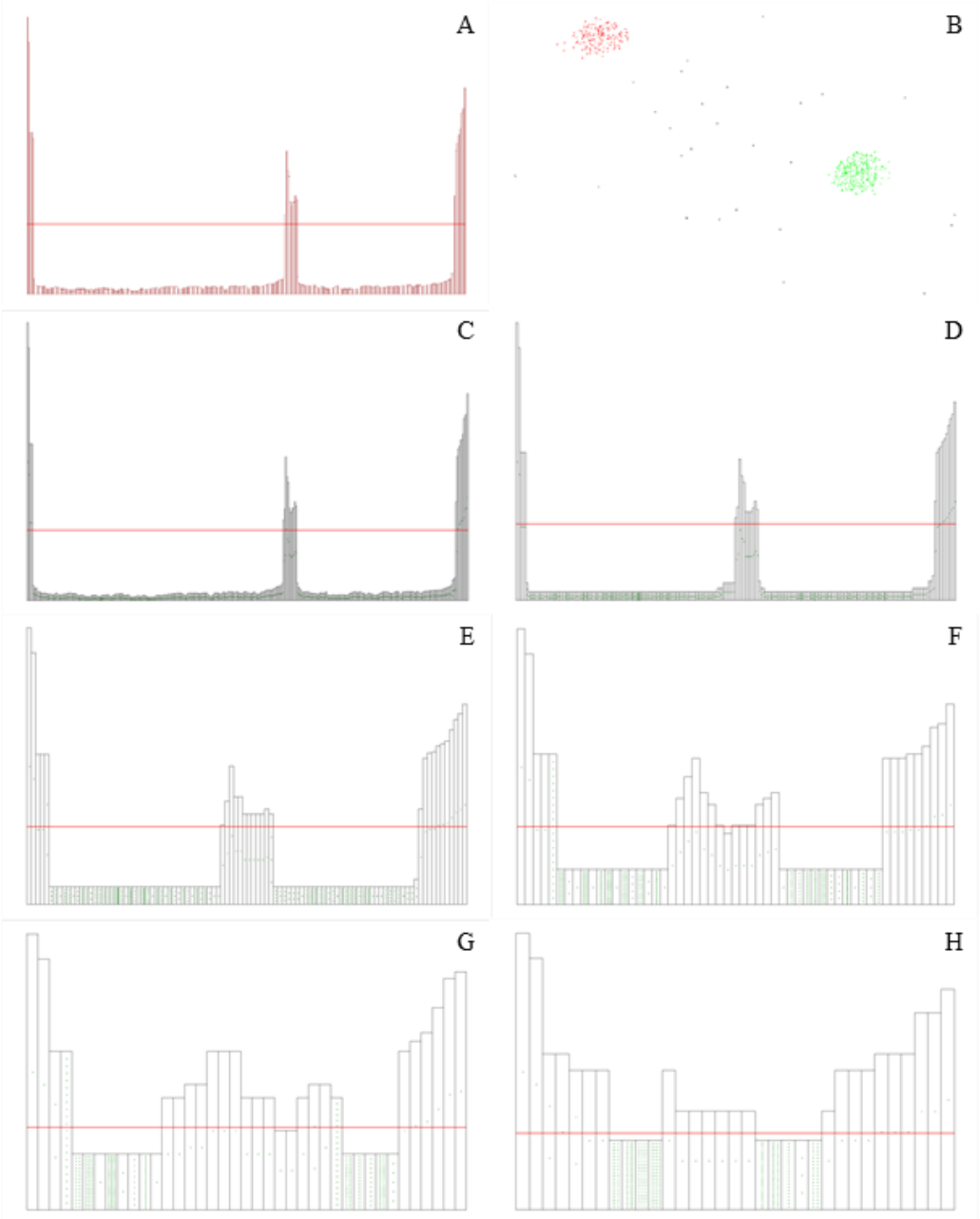

Figure 3: Results executed on PointSet500 data set with $\epsilon = 500$, $MinPts = 5$

In case of this point set the $\tau = 40$ is very high because it means that there are 10-12 gridlines in each dimension. This can cause inaccuracy, namely Figure 4 shows its result, where you see two points in two red circles which should be noise instead of clustered points.

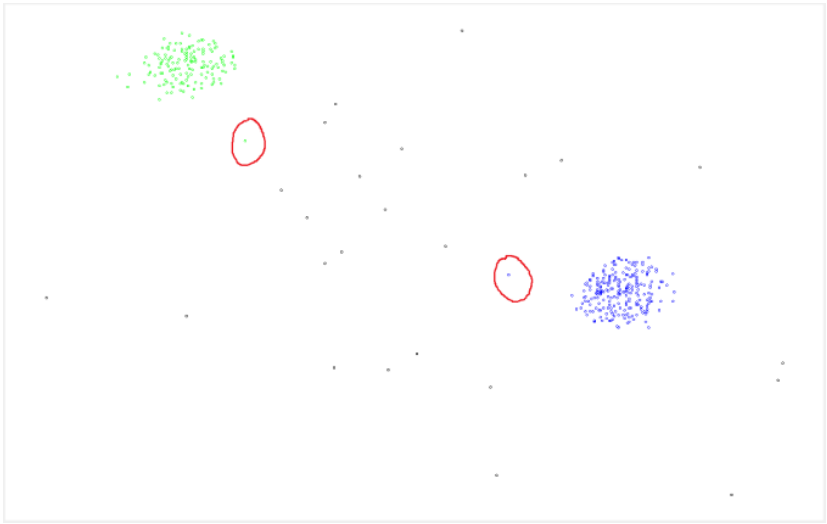

Figure 4: The clustered points of the PointSet500 data set executed with $\tau = 40$

The GridOPTICS with the other τ values created the same clusters as the OPTICS in the above described cases.

Figure 5 and Figure 6 show reachability plots of PointSet500 data set.

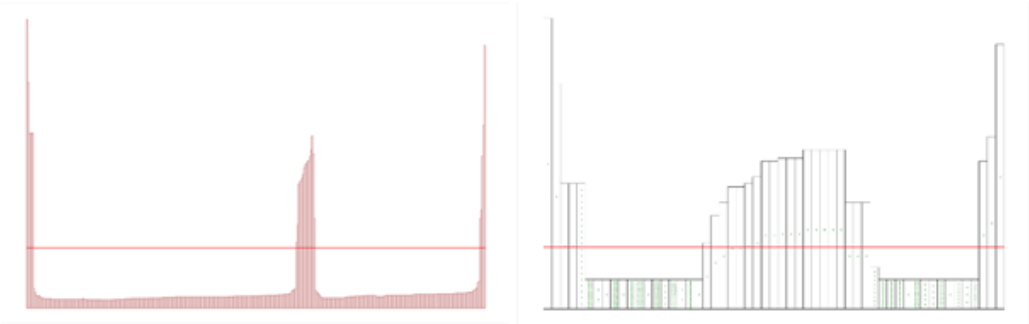

Figure 5: The reachability plots of the OPTICS (left side) and the GridOPTICS (right side) on the PointSet500 data set with $\epsilon = 500$, $MinPts = 20$, $\tau = 20$, and $\varphi = 42$

The GridOPTICS can cause inaccuracy because the original input points are substituted with grid points. The higher the τ is, the more inaccurate the result is. This inaccuracy can cause that a few input points are clustered but they

should be noise, or the GridOPTICS consider some input points to be noise but they should be clustered. In the most cases, these input points are in the border of a cluster. Another inaccuracy can happen when two or more clusters are contracted.

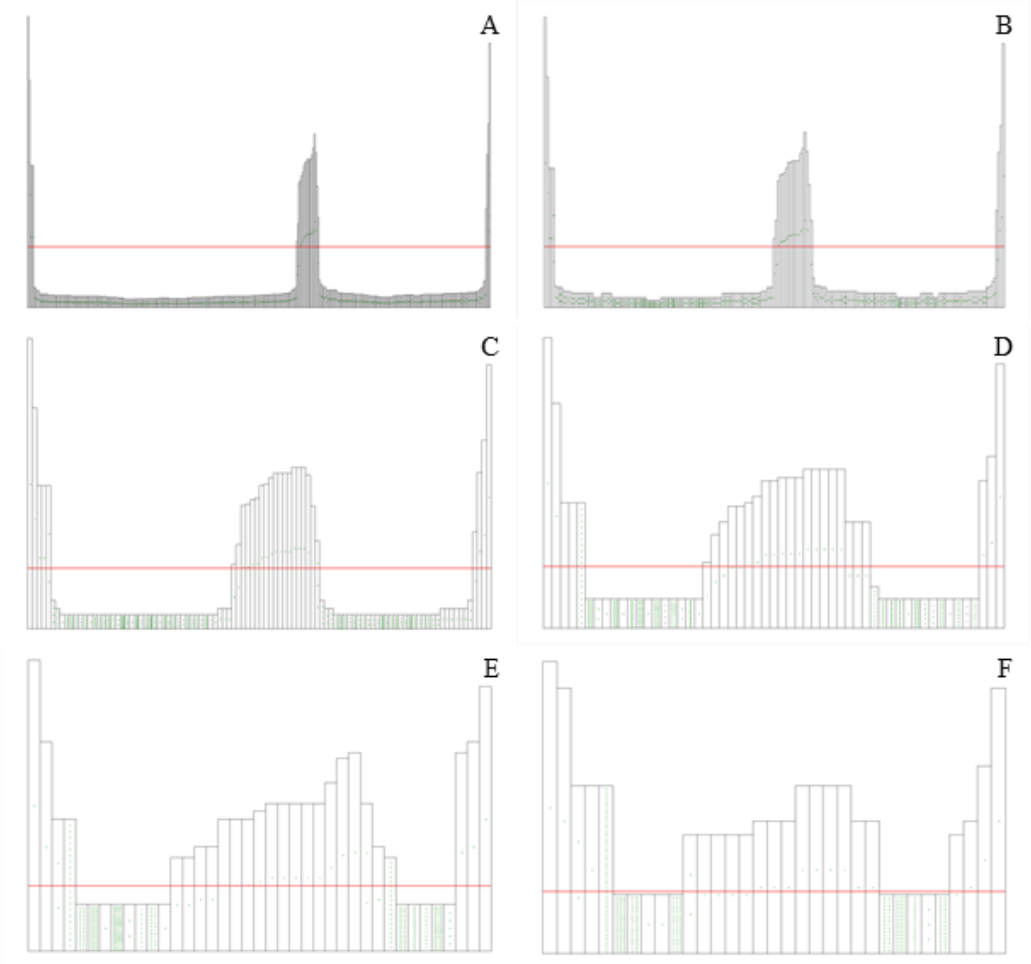

Figure 6: Results of the GridOPTICS on the PointSet500 data set with $\epsilon = 500$, $MinPts = 20$, $\varphi = 42$, and $\tau = 1, 5, 10, 20, 30$, and 40

The grid structure may be moved with τ' $(-\tau < \tau' < \tau)$ in a dimension but it may cause the same inaccuracy on an other side of the clusters.

Let us see other input points. PointSet1000 data set has 919 synthetic points, the x coordinates range between 234 and 572, whereas y coordinates are between 390 and 783. Table 4 shows the execution time of the GridOPTICS and the OPTICS on the PointSet1000 data set.

With higher $MinPts$ value, the reachability plots of the OPTICS are also plainer, there are not so many cuts in the valleys. In this way, the reachability plots of the GridOPTICS can be more similar to it. We can state if the $MinPts$ is higher, the GridOPTICS will be less or not inaccurate.

ϵ	MP	OT	τ	1	5	10	20	30
			NGP	863	484	241	118	76
400	5	64330ms	GOT	78660ms	13969ms	1849ms	266ms	86ms
500	10	69599ms	GOT	82527ms	14987ms	1648ms	290ms	64ms
400	20	61515ms	GOT	87628ms	16361ms	1759ms	245ms	97ms

Table 4: The execution time of the algorithms on the PointSet1000 data set

In case of this point set, there will be less than 10 gridline in each dimension if you choose $\tau \geq 30$. In this way, there will not be enough grid points in order that the GridOPTICS can give an accurate result.

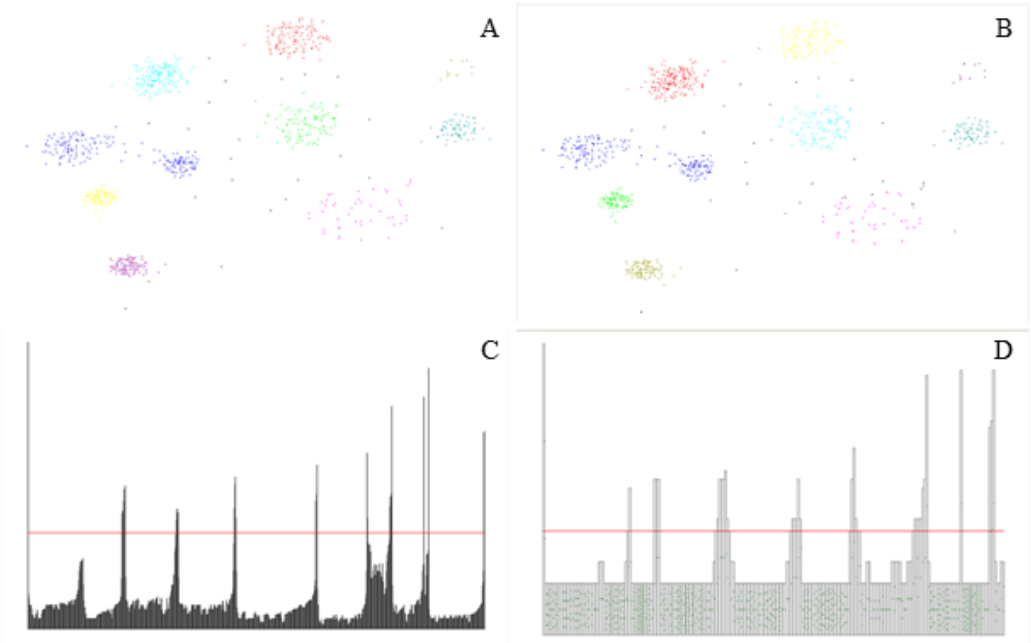

Figure 7: Results of the OPTICS and the GridOPTICS on the PointSet1000 data set with $\epsilon = 400$, $MinPts = 5$, $\tau = 10$, and $\varphi = 20$

Part A of Figure 7 shows clustered points and the C part shows the reachability plot resulted by the OPTICS with $\epsilon = 400$, $MinPts = 5$, and $\varphi = 20$. Subfigure B demonstrates clustered points, whereas Subfigure D displays the reachability plot of the GridOPTICS with $\epsilon = 400$, $MinPts = 5$, $\tau = 10$, and $\varphi = 20$. In

this case, the clusters are similar to each other, but on Subfigure B you can find inaccuracy, namely you can find input points which are noise instead of being clustered. These are caused by the transformation into the grid structure. At the same time, the GridOPTICS is thirty-four times faster than the OPTICS in this case.

PointSet4000 data set has 4028 synthetic points; the x coordinates range between 37 and 986, whereas y coordinates are between 20 and 933. Table 5 shows the execution time of the algorithms on this point set.

ϵ	MP	OT	τ	5	10	20	30
			NGP	2365	1120	486	309
1000	5	5474880ms	GOT	1518406ms	164218ms	14750ms	3334ms
1000	10	5699831ms	GOT	1679268ms	175580ms	14774ms	3998ms
1000	20	5429290ms	GOT	1483909ms	159955ms	13513ms	3542ms

Table 5: The execution time of the algorithms on the PointSet4000 data set

The OPTICS takes about 1,5 hour to give results for a set which has about 4000 points, whereas the GridOPTICS takes about 25 minute if $\tau = 5$, it takes 3 minutes if $\tau = 10$, and it takes less than a minute if $\tau \geq 20$.

Part A of Figure 8 shows the reachability plot, Subfigure C and E display the clustered points of the OPTICS with $\epsilon = 1000$, $MinPts = 20$, $\varphi = 25$ and $\varphi = 45$, in case of Subfigure C $\varphi = 25$, whereas on Subfigure E $\varphi = 45$. Subfigure B shows the reachability plot, whereas D ($\varphi = 25$) and F ($\varphi = 45$) parts show the clustered points of the GridOPTICS with $\epsilon = 1000$, $MinPts = 20$, $\tau = 20$, $\varphi = 25$ and $\varphi = 45$.

In this case, the GridOPTICS is faster about 400 times as the OPTICS but the GridOPTICS loses information, namely if $\varphi = 25$, the GridOPTICS finds a lot of noise (noise points are black on the figures), where the OPTICS finds clusters, furthermore, the GridOPTICS contracts clusters together. However, the GridOPTICS finds the main clusters similarly to the OPTICS. If $\varphi = 45$, the results of the two algorithms are almost the same.

ϵ	MP	OT	τ	5	10	20	30
			NGP	3282	1589	646	408
1000	5	10853077ms	GOT	4388269ms	466100ms	32677ms	8193ms

Table 6: The execution time of the algorithms on the PointSet5000 data set

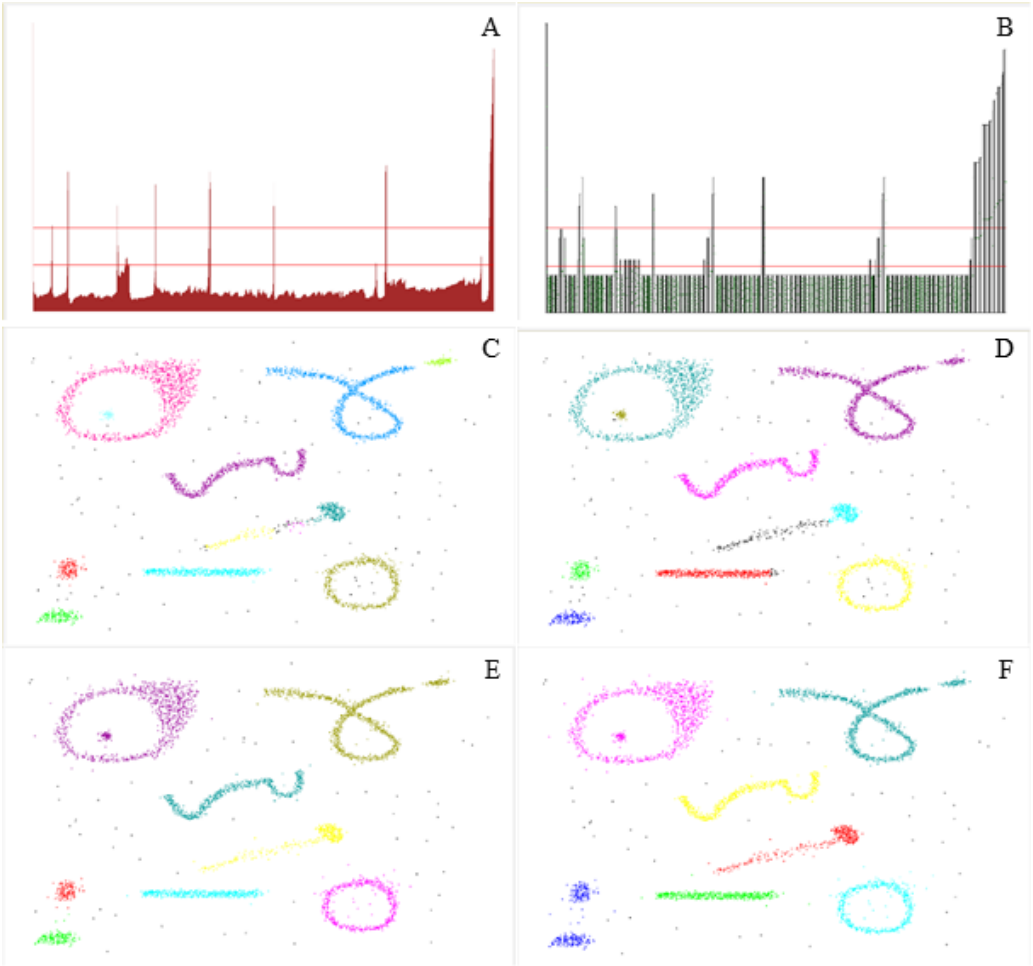

Figure 8: Results of the OPTICS and the GridOPTICS on the PointSet4000 data set with $\epsilon = 1000$, $MinPts = 20$, $\tau = 20$, $\varphi = 25$, and $\varphi = 45$

PointSet5000 data set has 5045 synthetic points, the x coordinates range between 22 and 978, whereas y coordinates are between 16 and 934. Table 6 shows the execution time of the algorithms on this point set.

Considering the execution time it took the OPTICS algorithm about three hours to give results, whereas it took the GridOPTICS about 7 minutes when $\tau = 10$.

Figure 9 shows the results executed on the PointSet5000 data set with $\epsilon = 1000$ and $MinPts = 5$. Subfigure A shows the reachability plot resulted by the OPTICS and Subfigure B shows the reachability plot resulted by the GridOPTICS with $\tau = 10$. $\varphi = 12$ and $\varphi = 32$ which values are represented by the red lines on these two subfigures. Subfigure C and E show the clustered points resulted by the OPTICS, in case of the C $\varphi = 12$, whereas in case of the E $\varphi = 32$. Similarly,

Subfigures D and F show the clustered points resulted by the GridOPTICS, in case of the D $\varphi = 12$, whereas in case of the F $\varphi = 32$.

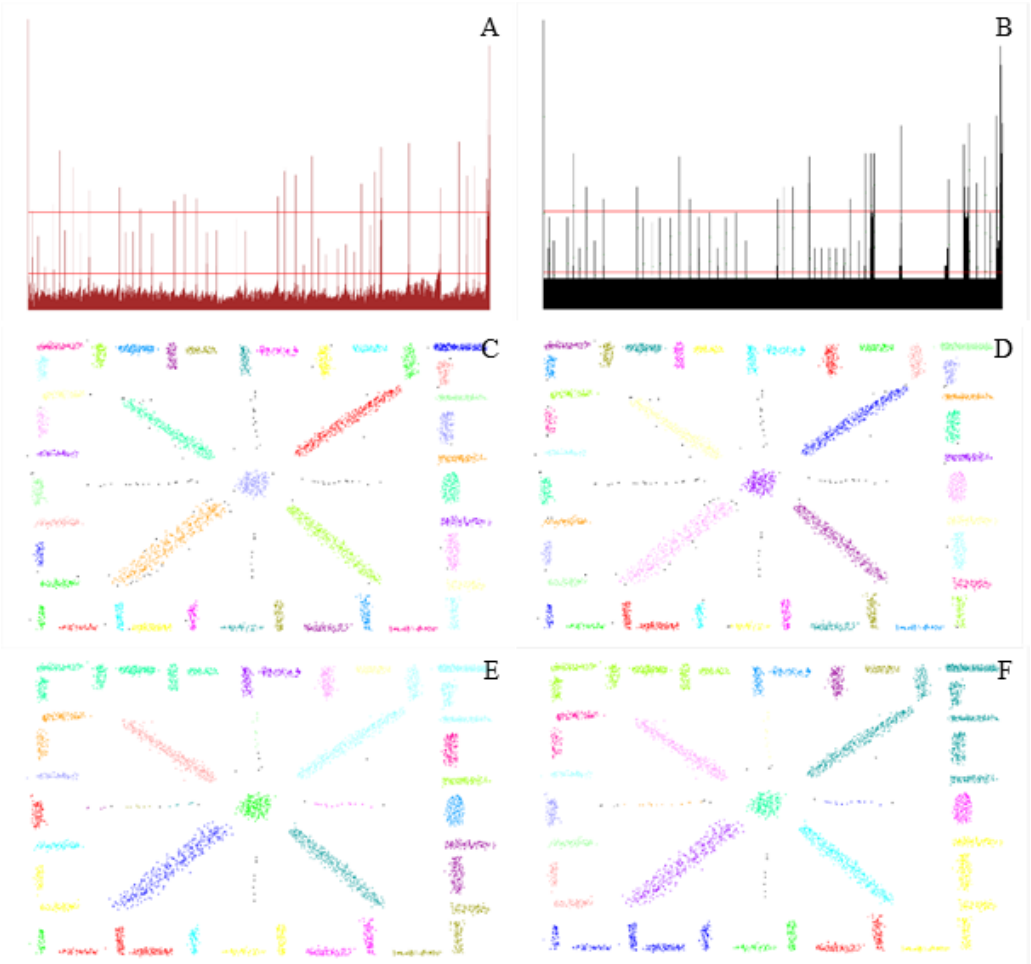

Figure 9: Results of the OPTICS and the GridOPTICS on the PointSet5000 data set with $\epsilon = 1000$, $MinPts = 5$, $\tau = 10$, $\varphi = 12$ and $\varphi = 32$

You can see a similar inaccuracy on Figure 9 as Figure 8, namely the GridOPTICS finds a lot of noise, where the OPTICS finds clusters, and the GridOPTICS contracts clusters together. At the same time, the execution time of the GridOPTICS is 23 times faster than the OPTICS in this case.

ϵ	MP	OT	τ	5	10	20
			NGP	1666	953	434
800	5	2154836ms	GOT	550441ms	96625ms	10150ms

Table 7: The execution time of the algorithms on the PointSet3000 data set

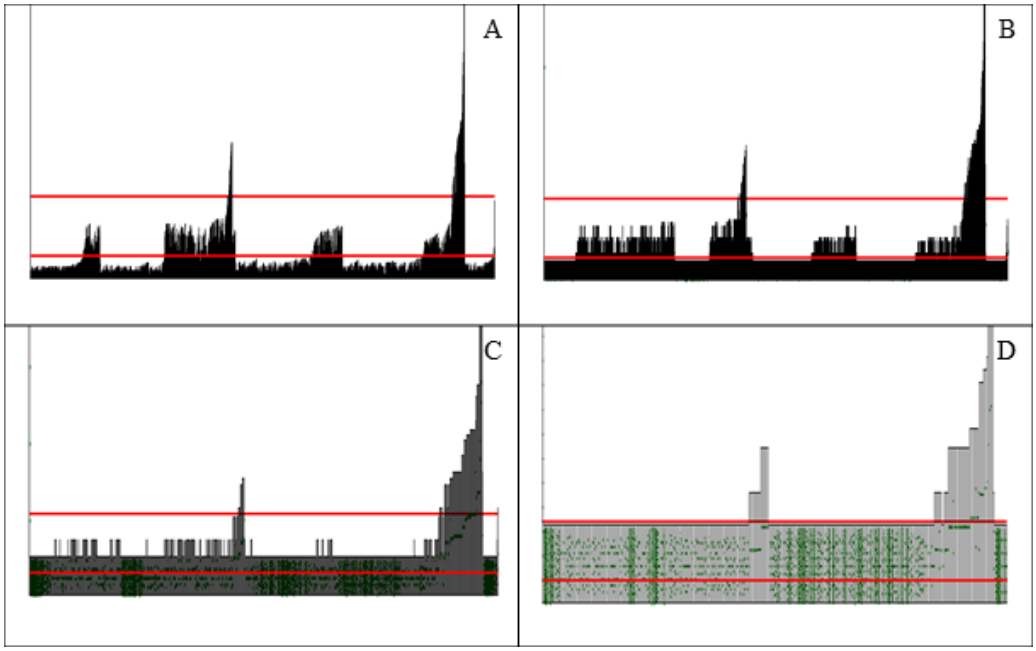

Figure 10: The reachability plots of the OPTICS and the GridOPTICS on the PointSet3000 data set with $\epsilon = 800$, $MinPts = 5$, $\tau = 5, 10$, and 20, $\varphi = 6$ and $\varphi = 21$

PointSet3000 data set has 2976 synthetic points; the x coordinates range between 204 and 877, whereas y coordinates are between 60 and 876. Table 7 shows the execution time of the algorithms on this point set.

Figure 10 shows the results executed on PointSet3000 data set with $\epsilon = 800$, $MinPts = 5$. Subfigure A shows the reachability plot resulted by the OPTICS, whereas B, C, and D parts show the reachability plots resulted by the GridOPTICS with $\tau = 5, 10$, and 20. $\varphi = 6$ and $\varphi = 21$ are chosen which are represented by the red lines on each subfigure. Subfigure A, B, C, and D are cut and enlarged in order that you can see the important parts of the reachability plots. The E and G parts of Figure 11 show the clustered points resulted by the OPTICS in case of the E $\varphi = 6$, whereas in case of the G $\varphi = 21$. Similarly, F and H parts of the figure show the clustered points resulted by the GridOPTICS $\tau = 5$, in case of the F $\varphi = 6$, whereas in case of the H $\varphi = 21$. Subfigure I and J show the clustered points resulted by the GridOPTICS, in case of I $\tau = 10$ and $\varphi = 21$, whereas in case of J $\tau = 20$ and $\varphi = 21$. In case of the GridOPTICS with $\tau = 10$ and 20 and $\varphi = 6$, all input points are noises.

You can see on Figure 10 and Figure 11 that it is not worth choosing higher value for the τ as 10 in this case, because the clusters having small granularity are lost.

Therefore, I can suggest using lower τ value, namely $\tau = 5$. The cardinality of the point set is slight, consequently the execution time is not so long.

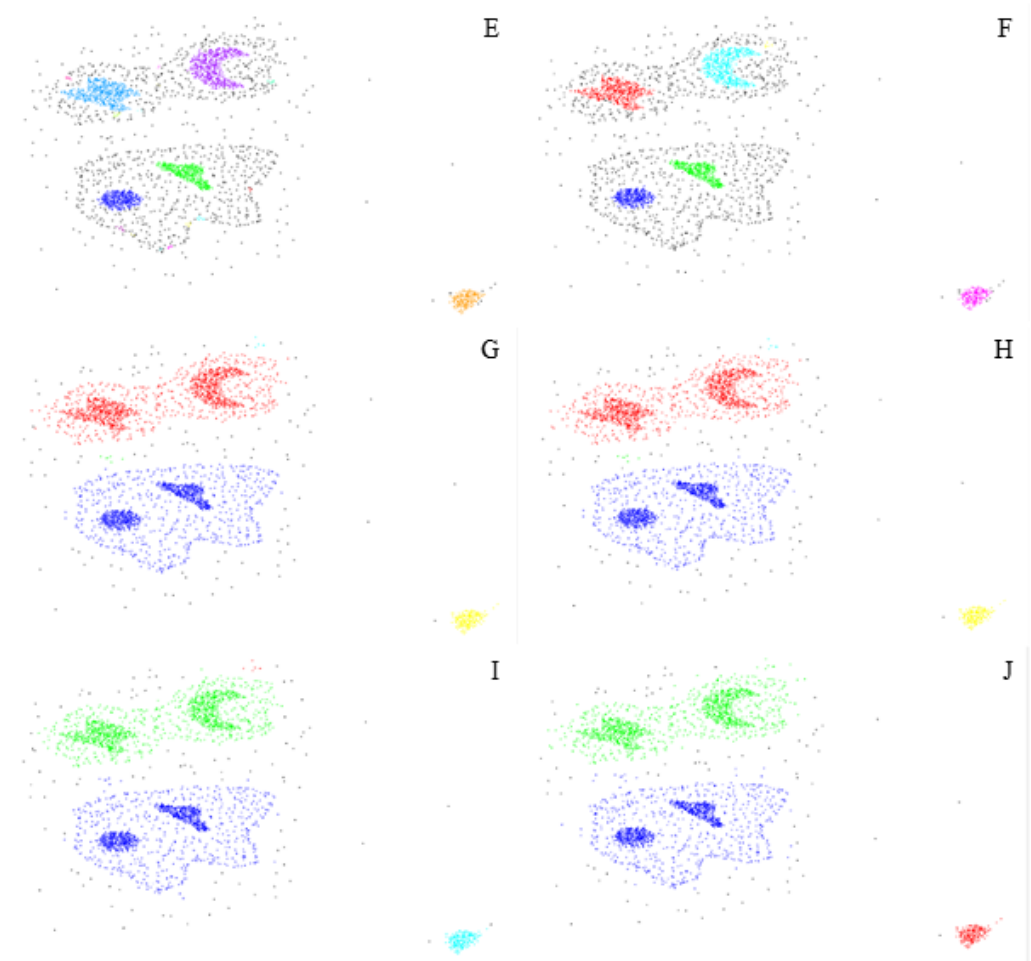

Figure 11: The clustered points of the OPTICS and the GridOPTICS on the PointSet3000 data set with $\epsilon = 800$, $MinPts = 5$, $\tau = 5, 10$, and 20, $\varphi = 6$ and $\varphi = 21$

ϵ	MP	OT	τ	110
			NGP	399
3000	5	12696ms	GOT	2237ms

Table 8: The execution time of the algorithms on the Aggregation data set

I also executed both algorithms on some clustering data sets from the http://cs.joensuu.fi/sipu/datasets/ website. Figure 12 shows the results executed on the Aggregation data set (Gionis, 2007), where Table 8 shows the execution

time of the algorithms. The Aggregation data set has 788 points, the x coordinates range between 335 and 3655, whereas y coordinates are between 195 and 2915.

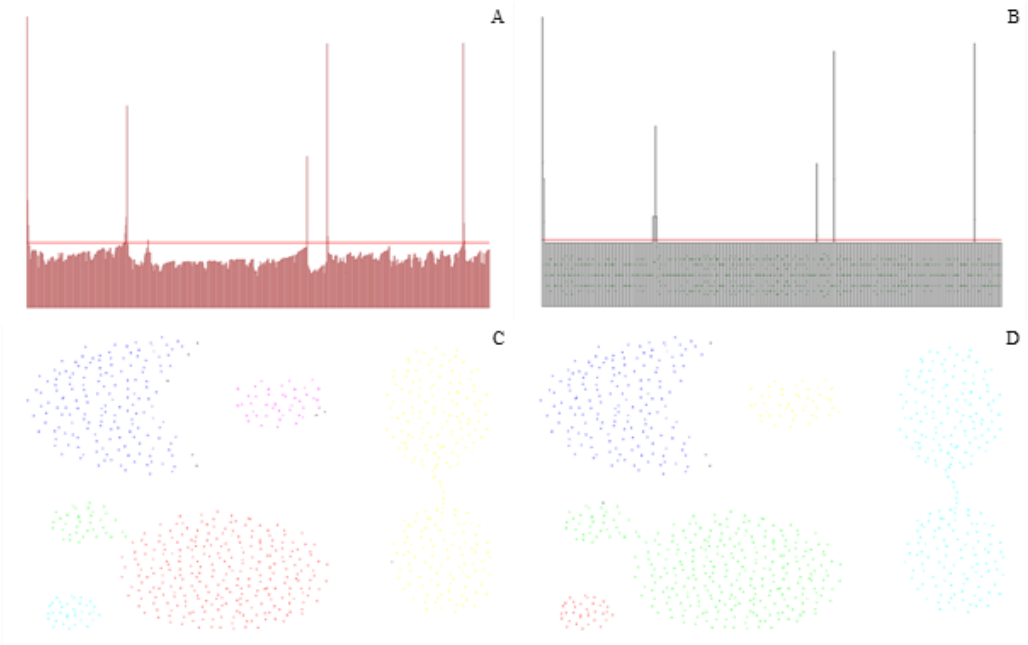

Figure 12: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the Aggregation with $\epsilon = 3000$, $MinPts = 5$, $\tau = 110$ and $\varphi = 115$

Figure 13 shows the results executed on the Dim2 data set, where Table 9 shows the execution time of the algorithms. The Dim2 data set has 1351 points, the x coordinates range between 0 and 978207, whereas y coordinates are between 0 and 1000000.

ϵ	MP	OT	τ	10000
			NGP	145
1000000	5	70651ms	GOT	212ms

Table 9: The execution time of the algorithms on the Dim2 data set

Figure 14 shows the results executed on the A1 data set (Kärkkäinen and Fränti, 2002), where Table 10 shows the execution time of the algorithms. The A1 data set has 3000 points, the x coordinates range between 0 and 65535, whereas y coordinates are between 32064 and 64978.

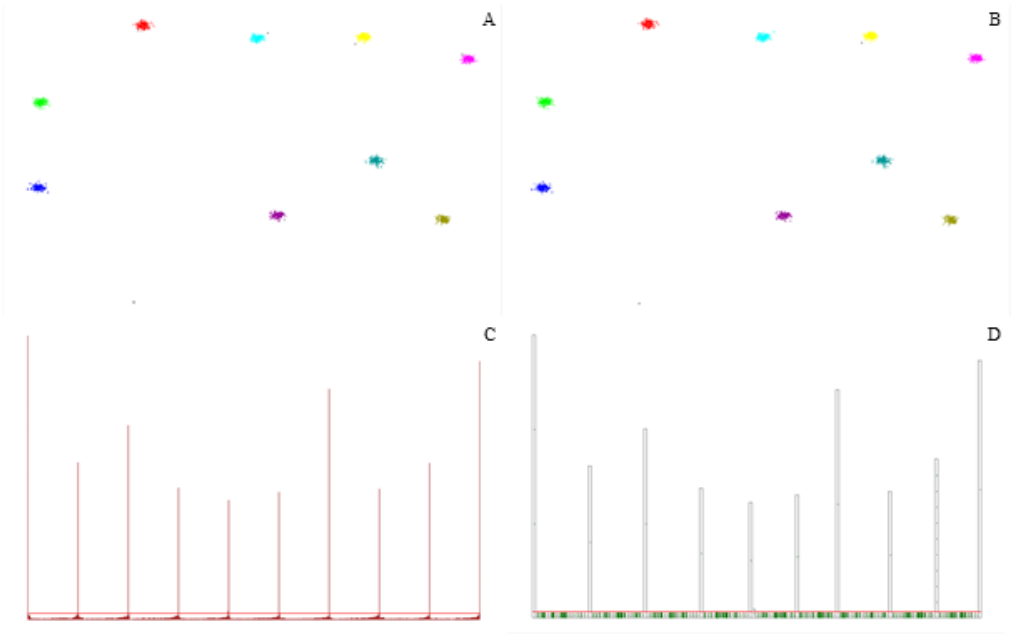

Figure 13: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the Dim2 with $\epsilon = 1000000$, $MinPts = 5$, $\tau = 10000$ and $\varphi = 10100$

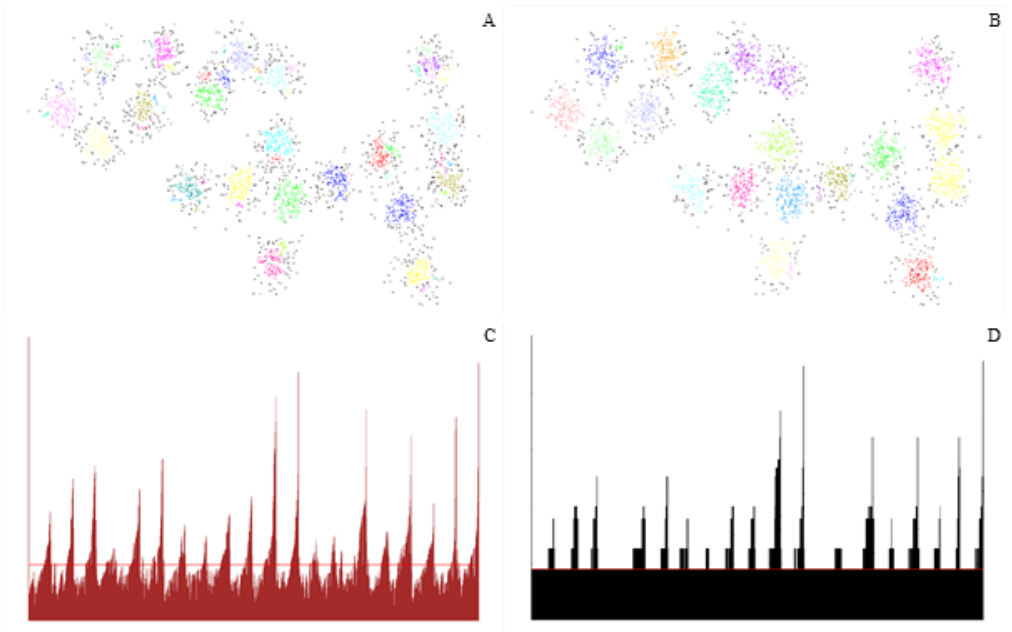

Figure 14: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the A1 with $\epsilon = 60000$, $MinPts = 5$, $\tau = 500$ and $\varphi = 501$

ϵ	MP	OT	τ	500
			NGP	1725
60000	5	697345ms	GOT	173274ms

Table 10: The execution time of the algorithms on the A1

Figure 15 shows the results executed on the S3 data set (Fränti and Virmajoki, 2006), where Table 11 shows the execution time of the algorithms. The S3 data set has 5000 points, the x coordinates range between 32710 and 942327, whereas y coordinates are between 70003 and 947322.

ϵ	MP	OT	τ	10000
			NGP	2530
100000	5	3226123ms	GOT	540397ms

Table 11: The execution time of the algorithms on the S3

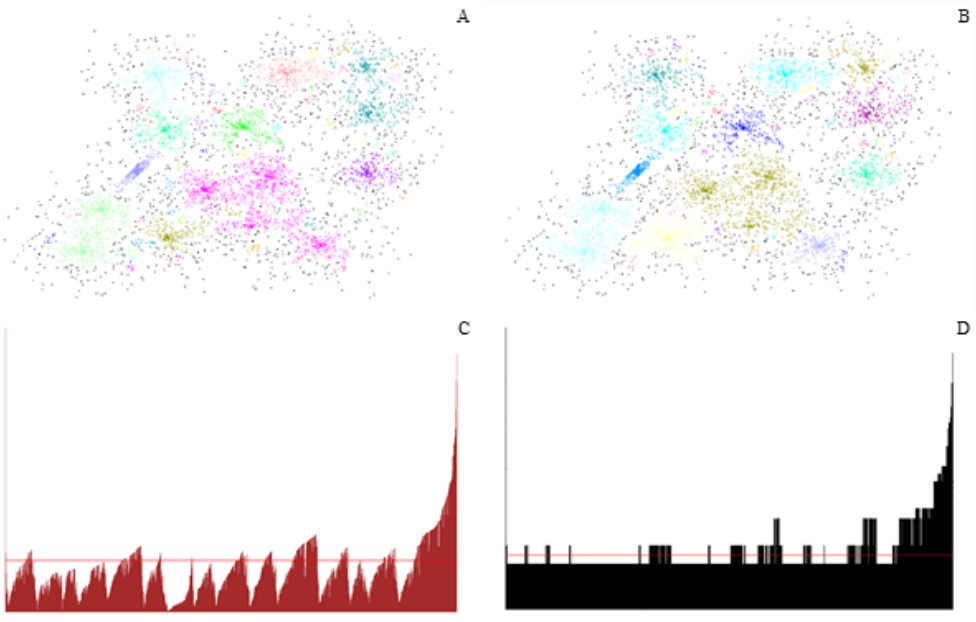

Figure 15: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the S3 with $\epsilon = 100000$, $MinPts = 5$, $\tau = 10000$, and $\varphi = 12000$

Figure 16 shows the results executed on the Unbalance data set, where Table 12 shows the execution time of the algorithms. The Unbalance data set has 6500

points, the x coordinates range between 139779 and 575805, whereas y coordinates are between 271530 and 440940.

ϵ	MP	OT	τ	4000
			NGP	378
500000	5	7091022ms	GOT	1997ms

Table 12: The execution time of the algorithms on the Unbalance data set

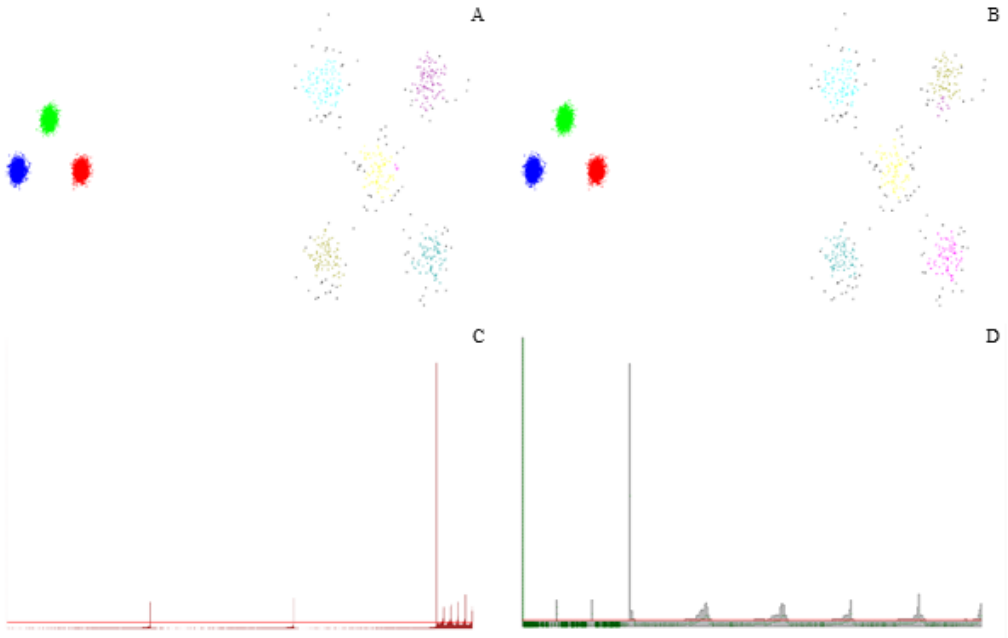

Figure 16: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the Unbalance data set with $\epsilon = 500000$, $MinPts = 5$, $\tau = 4000$, and $\varphi = 5000$

Figure 17 shows the results executed on the t4.8k data set (Karypis et al., 1999), where Table 13 shows the execution time of the algorithms. The t4.8k data set has 8000 points, the x coordinates range between 14642 and 634957, whereas y coordinates are between 21381 and 320874.

ϵ	MP	OT	τ	5500
			NGP	3037
600000	5	13664865ms	GOT	947071ms

Table 13: The execution time of the algorithms on the t4.8k

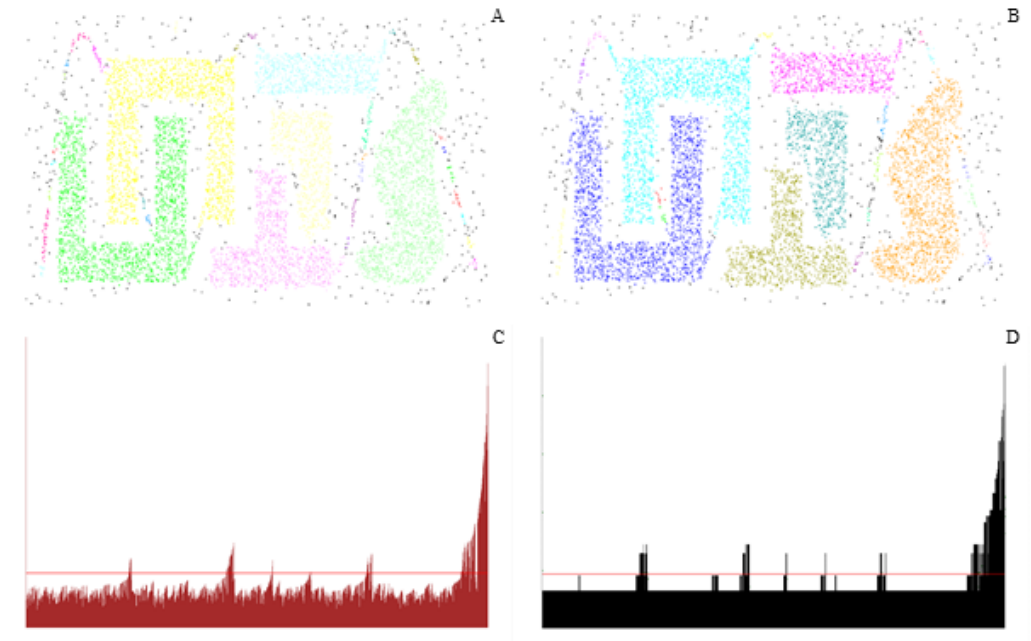

Figure 17: The reachability plots and clustered points of the OPTICS (A, C) and the GridOPTICS (B, D) on the t4.8k with $\epsilon = 600000$, $MinPts = 5$, $\tau = 5500$, and $\varphi = 5800$

Figure 18, Figure 19, and Figure 20 show examples for the results of the GridOPTICS executed on data sets with 100000 and 50000 data points.

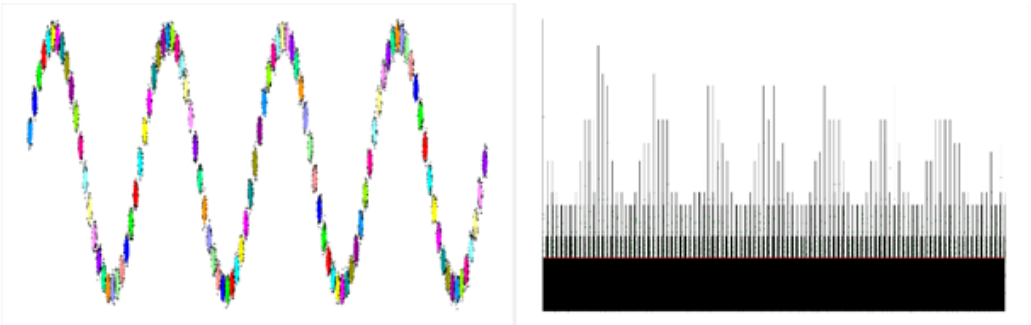

Figure 18: The clustered points and the reachability plot resulted by the GridOPTICS on BIRCH2

Figure 18 shows the clustered points and the reachability plot resulted by the GridOPTICS with $\epsilon = 10000$, $MinPts = 50$, $\tau = 1000$, and $\varphi = 1010$ on BIRCH2 data set (Zhang et al., 1997), whose cardinality is 100000. The execution time is 1279471 milliseconds. The number of the grid points is 7131. The x coordinates

range between 47734 and 1000000, whereas the y coordinates range between 0 and 86244.

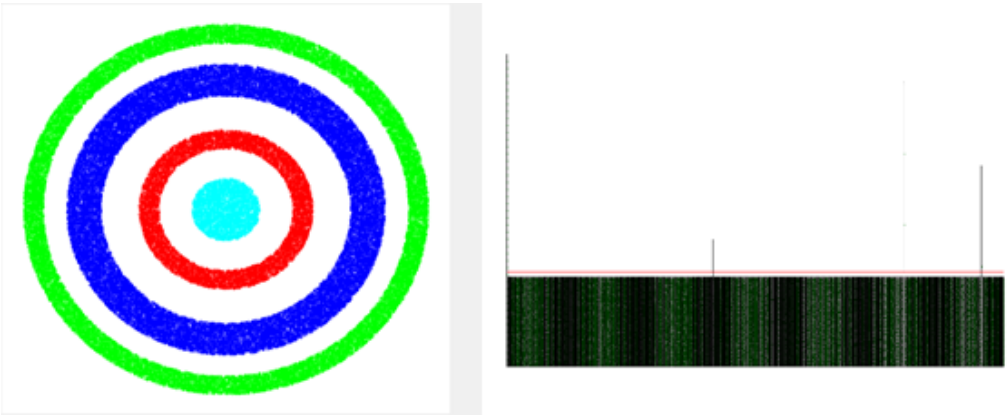

Figure 19: The clustered points and the reachability plot resulted by the GridOPTICS on PointsetCircle50000 synthetic data set

Figure 19 shows the clustered points and the reachability plot resulted by the GridOPTICS with $\epsilon = 800$, $MinPts = 5$, $\tau = 200$, and $\varphi = 21$ on PointsetCircle50000 synthetic data set, whose cardinality is 49968. The execution time is 121626 milliseconds. The number of the grid points is 1023. The x coordinates range between 81 and 920, whereas the y coordinates range between 81 and 920.

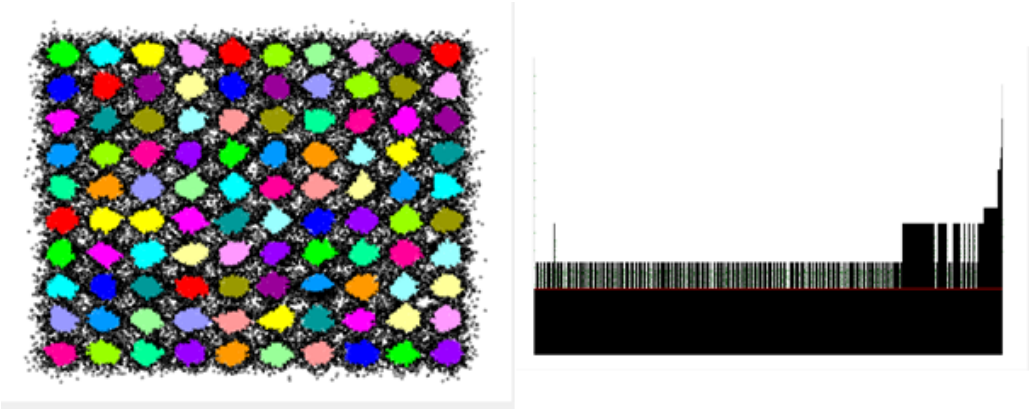

Figure 20: The clustered points and the reachability plot resulted by the GridOPTICS on BIRCH1 data set

Figure 20 shows the clustered points and the reachability plot resulted by the GridOPTICS with $\epsilon = 100000$, $MinPts = 50$, $\tau = 8000$, and $\varphi = 8010$ on BIRCH1 data set (Zhang et al., 1997), whose cardinality is 100000. The execution

time is 191413781 milliseconds. The number of the grid points is 13507. The x coordinates range between 1371 and 996108 whereas the y coordinates range between 0 and 1000000.

2.4.6 Application of the GridOPTICS

I suggest using the GridOPTICS if you have more than 500-600 input points, because if you have only less than 500 input points, the OPTICS works fast, it takes only 1 second to give results. But, you can read from Table 4 (PointSet1000) that if you have about 1000 points, it takes the OPTICS about 1 hour to give results on a simple PC, whereas it takes the GridOPTICS 13 second if $\tau = 5$.

Moreover, the GridOPTICS is very useful, if you have a data set where a lot of input points have the same coordinates with each other, which means that the number of the grid points may be a fractional part of the number of input points even if $\tau = 1$. Figure 21 shows an example for such a data set, where $\tau = 1$ and the cardinality of the grid points is fewer with one order of magnitude as the cardinality of input points.

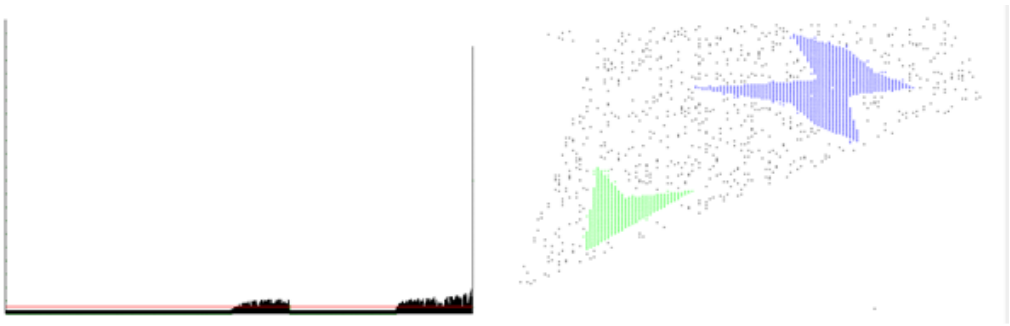

Figure 21: Results of the GridOPTICS on the PointSet41000 data set

There are the results of the GridOPTICS with $\epsilon = 200$, $MinPts = 5$, $\tau = 1$, and $\varphi = 2$ on the PointSet41000 data set, whose cardinality is 41000 on Figure 21. The cardinality of the grid points is 2541. The execution time is 1925881 millisecond. The x coordinates range between 27 and 158 whereas the y coordinates range between 14 and 199.

In most examples, I used large ϵ values in order to show entire reachability plots. Of course, you can execute the GridOPTICS with smaller ϵ, which will results that it may not find the rough-grained clusters. The OPTICS works similarly. In Figure 22 you can find an example for smaller ϵ, namely they are the reachability plot and the clustered points of the GridOPTICS on PointSet5000 data set with $\epsilon = 40$, $MinPts = 5$, $\tau = 10$, and $\varphi = 32$.

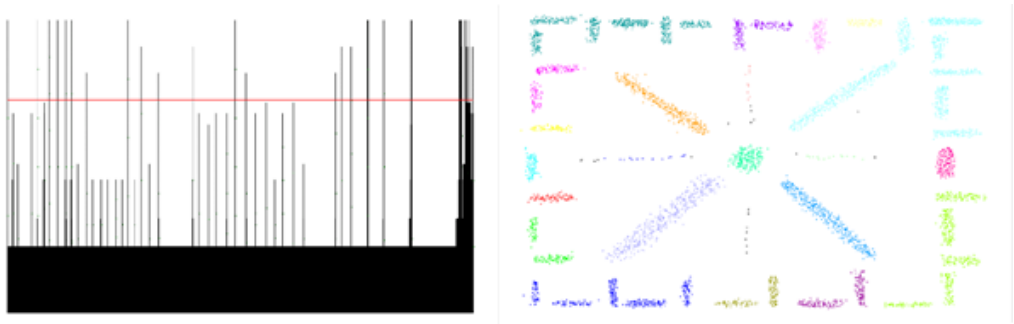

Figure 22: Results of the GridOPTICS on the PointSet5000 data set

The τ should be less or equal with ϵ, otherwise a cluster will consist of the input points which belong to a grid point. Of course, it can also be considered a clustering algorithm.

I would estimate the value of τ based on the cardinality of the point set and the range of the coordinates. I would choose the τ in order to be at least 500-1000 grid points. If there are less than 1000 grid points, the GridOPTICS gives results in a few minute. However, 500-1000 grid points are not enough for large data sets. Hence, I could consider the relationship between the execution time and the cardinality of the grid points, because the execution time substantially depends on the cardinality of the grid points. Finally, who will apply this algorithm, should decide what they need. On the one hand, you can set a higher value to τ, consequently the algorithm will be faster but less accurate, however, in case of large data sets the high τ values may scarcely cause accuracy problems. On the other hand, you can choose lower value for τ, which will result in longer execution time, but more accurate results.

2.4.7 Future work

I plan to try out the algorithm for high dimensions and with more types of distances. One type of real world examples can be that the input data are GPS coordinates, which need a special distance.

I also plan to find an appropriate method to measure the quality of the GridOPTICS algorithm considering the OPTICS. Both algorithms generate reachability plots and it would be better if we could compare the reachability plots instead of the resulted clusters. But, I cannot find any quality measurement technique for the reachability plots. I plan to give one in the future.

3 Biomedical signal processing

Living creatures are constantly generating data about the activity of their body. This data can be captured and can be processed. As a result, some information of a living creature can be measured and recognized. The medicine likes to use the signal processing in many parts of the diagnostic procedure. Simpler methods are the cases when the measurement gives information about the patient at a specific point in time, such as blood glucose level, heart rate, blood pressure or oxygen saturation measurements. Lots of measurements of a patient and their time can be plotted on a chart in order that physicians can make a better diagnosis. Other methods can give more information of the human based on monitoring the activity of the body parts, such as brain, muscle, and heart. In this way, physicians can find the abnormal activities, and based on this information they make the diagnosis. (Sörnmo and Laguna, 2005)

Monitoring of patients at home circumstances is becoming more and more popular in health care, since biomedical signal processing offers non-invasive techniques, which are safe and comfortable to perform it. The patients get traditional bio-measurement tools, data of which are recorded and transferred to a server where they are processed by software. The physicians get a lot of information about the patient, but the algorithms and the mathematical formulas of the software can help the physician to find the activity which differs from the normal activity. (Sörnmo and Laguna, 2005)

Nowadays the technology enables real-time monitoring, which helps recognize diseases and adverse events earlier and manage chronic diseases.

3.1 Cardiospy software of Labtech Ltd.

The Hungarian company, Labtech Ltd. deals with developing and manufacturing PC-based ECG Systems. Its products are used for monitoring and analyzing the heart's activity and its electronic signals, these are mainly ambulatory (Holter) ECGs and Resting and Stress Test ECG Systems. The company manufactures the products and develops their analysis software called Cardiospy, which works on Windows operating system. In newer developments, the company provides telemedicine solutions for Android-based tablets or smartphones.

Its products are used in more countries, such as Japan, Germany, France, Turkey, Spain, The Netherlands, Romania, Ukraine, Russia, Iran, Algeria, Poland, Austria, and The Philippines. (Labtech, 2015)

3.2 ECG – Electrocardiogram

An electrocardiogram (ECG) describes the electrical activity of the heart recorded by electrodes placed on the body surface. The electrodes detect tiny electrical changes on the skin caused by the action potentials of the cardiac cells during the cells contract. The voltages of the electrodes are recorded regularly during a time interval and they are plotted in the ECG. (Abdulla, 2015), (Sörnmo and Laguna, 2005)

The ECG provides important information about the patient's heart rhythm, a previous heart attack, increased thickness of heart muscle, signs of decreased oxygen delivery to the heart, problems with conduction of the electrical current from one part of the heart to another, and problems with blood delivery to the heart muscle. (Abdulla, 2015)

The simply ECG test lasts for 5 to 10 minutes. A stress ECG test is when measurements are taken when the subject patient is exercising until a target heart rate is achieved. A stress test would have a higher probably of identifying issues with the heart. (AmperorDirect, 2015)

During a Holter ECG or ambulatory ECG test the patient is monitored up to 24 hours. They wear a portable device which records the signs of the electrodes. The long recordings are useful for recognizing occasional cardiac arrhythmias or epileptic events, which can hardly be found with short ECG tests.

3.2.1 ECG channels

The number and position of electrodes varies by ECG models. The ECG devices whose task is to make recordings during a short period usually have 10 electrodes, from which 12 leads can be derived. The devices can plot ECG signs with 1, 2, 3, 6, or 12 channels. To be comfortable, Holter monitors use less electrodes. (Sörnmo and Laguna, 2005)

The Labtech Ltd. offers Holter ECG devices with 3, 5, 7 or 12 electrodes, which can plot ECG signs with 1, 2, 3 or 12 channels. The number of the channels depends on the number of the electrodes. Cardiospy plots ECG signs with 1, 2 or 3 channels. In the documentation, the company introduces how the electrodes have to be placed on a patient. There are more than one opportunities how the electrodes have to be placed. Figure 23 presents only an example for a device with 7 electrodes. The figure also presents the relationship between the electrodes and the channels, namely the colored numbers show the channels. Beside the colored

numbers, there are positive or negative signs, which show that the channels are bipolar, namely the ECG signs can be mirrored if the electrodes are replaced. The black N shows the place of the neutral electrode. (Labtech, Electrode placement, 2015)

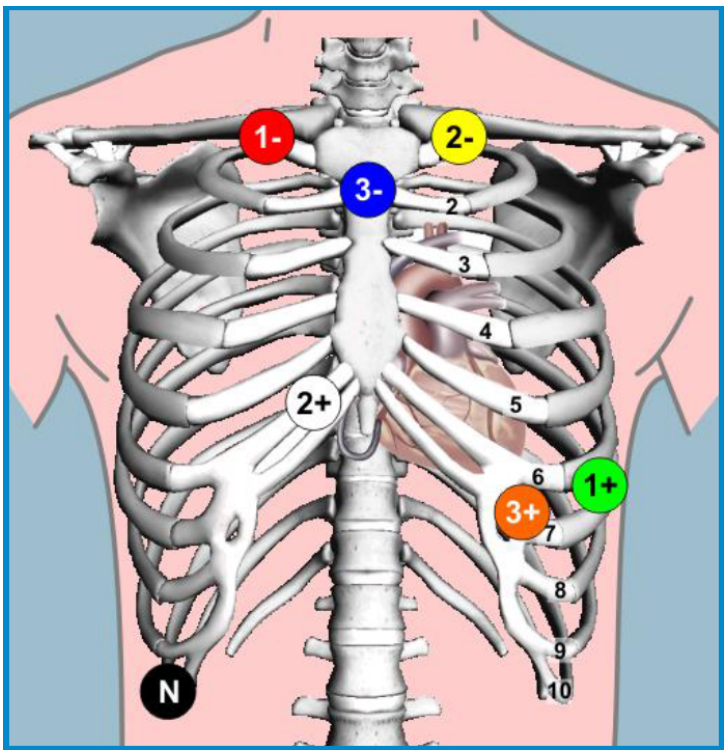

Figure 23: The places of the electrodes of ECG device with channel numbers

The Holter ECG devices of Labtech Ltd. can produce various recording rates, namely 128, 256, 512 or 1024 Hz. (Labtech, 2015)

3.2.2 ECG waves

Figure 24 shows an ECG sign. The main waves on the ECG are given the names P, Q, R, S, T, and U. Each wave represents depolarization ('electrical discharging') or repolarization ('electrical recharging') of a certain region of the heart. (Houghton and Gray, 2008)

In the case of a normal heart, each beat begins with the discharge ('depolarization') of the sinoatrial (SA) node (see Figure 25), high up in the right atrium. This is a spontaneous event and it does not cause any noticeable wave on the standard ECG.

The first detectable wave appears when the impulse spreads from the SA node to depolarize the atrium, which produces the *P wave* of the ECG. The atrium contains relatively little muscle, so the voltage generated by atrial depolarization is relatively small. The P wave will be a positive (upward) deflection.

After flowing through the atrium, the electrical impulse reaches and activates the atrioventricular (AV) node, which is located low in the right atrium. Its activation does not produce an obvious wave on the ECG, but it does contribute to the time interval between the P wave and the subsequent Q or R wave.

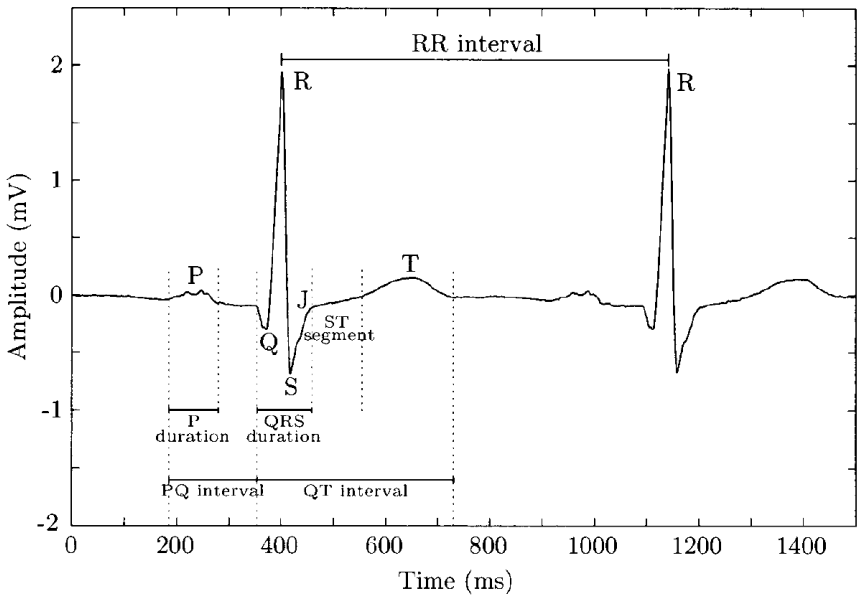

Figure 24: An ECG wave (Sörnmo and Laguna, 2005)

The *PR interval* is measured from the beginning of the P wave to the beginning of the R wave and it is normally between 0.12 s and 0.20 s.

Once the impulse has traversed the AV node, it enters the bundle of His, a specialized conducting pathway that passes into the interventricular septum and it divides into the left and right bundle branches. Current normally flows between the bundle branches in the interventricular septum, from left to right, and this is responsible for the first deflection of the *QRS complex*.

By convention, if the first deflection of the QRS complex is downward, it is called a *Q wave*. The first upward deflection is called an *R wave*, whether or not it follows a Q wave. A downward deflection after an R wave is called an *S wave*. A variety of complexes is possible.

The right bundle branch conducts the wave of depolarization to the right ventricle, whereas the left bundle branch divides into anterior and posterior fascicles that conduct the wave to the left ventricle. The conducting pathways end by dividing into Purkinje fibers that distribute the wave of depolarization rapidly throughout both ventricles. The depolarization of the ventricles, represented by the QRS complex, is normally complete within 0.12 s. QRS complexes are 'positive' or 'negative', depending on whether the R wave or the S wave is bigger. This, in turn, will depend on the view each lead has of the heart.

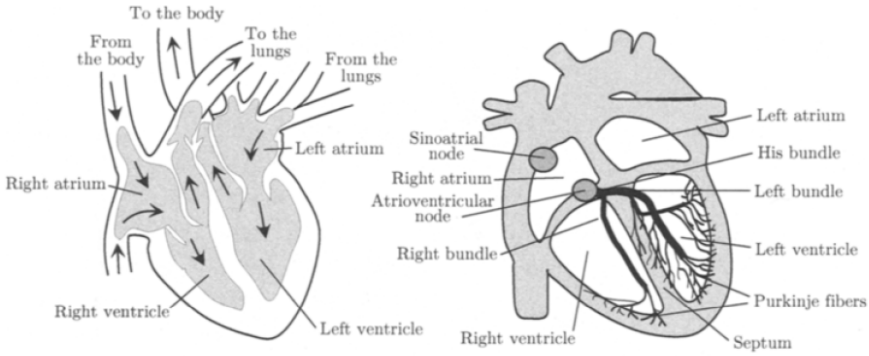

Figure 25: Parts of a heart (Sörnmo and Laguna, 2005))

The *ST segment* is the transient period in which no more electrical current can be passed through the myocardium. It is measured from the end of the S wave to the beginning of the T wave.

The *T wave* represents repolarization ('recharging') of the ventricular myocardium to its resting electrical state.

The *QT interval* measures the total time for activation of the ventricles and recovery to the normal resting state.

The origin of the *U wave* is uncertain, but it may represent repolarization of the interventricular septum or slow repolarization of the ventricles. U waves can be difficult to identify.

The *RR interval* is the duration of the ventricular cardiac cycle, whereas the *PP interval* is the duration of the atrial cycle. (Houghton and Gray, 2008)

The P wave cannot usually be recognized, because it is a very small sign. The QRS complex is much larger. The amplitude of a wave is measured with reference to the ECG baseline level, commonly defined by the isoelectric line which immediately

precedes the QRS complex. In most leads, the P wave has positive polarity and a smooth, monophasic morphology. Its amplitude is normally less than 300 μV, and its duration is less than 120 ms. The spectral characteristic of a normal P wave is usually considered to be low-frequency, below 10-15 Hz.

It is sometimes problematic to determine the time instants that define the beginning and end of a P wave because of a low amplitude and smooth morphology. As a result, the analysis of individual P waves is excluded from certain ECG applications where the presence of noise is considerable.

The QRS complex reflects depolarization of the right and left ventricles which lasts for about 70-110 ms in the normal heart. The QRS complex may be composed of less than three individual waves. The morphology of the QRS complex is highly variable and depends on the origin of the heartbeat. The QRS duration may extend up to 250 ms, and is sometimes composed of more than three waves. Since the QRS complex has the largest amplitude of the ECG waveforms, sometimes reaching 2-3 mV, it is the waveform of the ECG which is firstly identified in any type of computer-based analysis. Due to its steep slopes, the frequency content of the QRS complex is considerably higher than that of the other ECG waves and is mostly concentrated in the interval 10-50 Hz. The T wave extends about 300 ms after the QRS complex. The position of the T wave is strongly dependent on heart rate, becoming narrower and closer to the QRS complex at rapid rates. The normal T wave has a smooth, rounded morphology which, in most leads, is associated with a single positive peak.

At rapid heart rates, the P wave merges with the T wave, making the T wave end point become fuzzy as well as the P wave beginning. As a result, it becomes extremely difficult to determine the T wave end point because of the gradual transition from wave to baseline. (Sörnmo and Laguna, 2005)

3.2.3 Analysis of ECG by algorithms

The first important analysis of ECG is to find the QRS complex on the signal. The R-peak is the standard representation of ECG beats. In the QRS detection, the R-peak has to be found, which will be marked as an annotation of the heartbeat. It is not easy to find the R-peak, because the QRS complex waveform has a huge diversity. Pan and Tompkins (1985) give an early solution to this problem, but it is constantly being improved (Christov, 2004), (Chouhan and Mehta, 2008), (Elgendi et al., 2009), (Rani et al., 2015).

The next important analysis is to detect the type of a heartbeat. It is a more complex problem than QRS detection. The most important type of heartbeats

is the ventricular beats. But, it is hard to tell difference between ventricular and normal heartbeats, and ventricular and supraventricular heartbeats. There are many publications which try to determine the type of heartbeats (Dotsinsky and Stoyanov, 2004), (Cepek et al., 2007), (Tsipouras, 2005), (De Chazal et al., 2004), (Shahram and Nayebi, 2001), (Ye et al., 2012), (Dilmac and Korurek, 2015), (Martis et al., 2013), (Zidelmal et al., 2013).

Micó et al. (2005) cluster the ECG signs in order that the number of the heartbeat will be reduced, so physicians can make diagnosis faster and easier. First they use a compressing technique, namely a polygonal approximation to reduce the signal size. Then they extract the important features of a heartbeat. Finally, they use a clustering technique on the representations of the beats. They get a major group with normal heartbeats, and a few clusters, in which there are a few beats. The diagnostic maker only needs to analyze the latter clusters, in which there can be both outliers and abnormal heartbeats.

Zheng et al. (2007) also use clustering techniques in order to make the classification of the beats simpler.

Wavelet transformation has been playing an important role in ECG signal processing in the last few years.

There are many techniques to remove noise and artifacts from the ECG signals, such as digital filters, adaptive method and wavelet transform thresholding methods. ECG signal can be considered as a non-stationary signal, so it is not easy to denoise. Recently the wavelet transform has been proven to be a useful tool for non-stationary signal analysis. (Alfaouri and Daqrouq, 2008)

Discrete Wavelet Transform has been widely used in ECG signal analysis since this type of signals (namely short high-frequency impulses (R wave) which are superimposed on slowly varying waveforms (P, T and U waves), including artifacts and noise) favors processing tools with variable time-frequency resolution. "Discrete Wavelet Transform enables reliable identification of specific points and segments characterizing the ECG waveforms and it may also yield high compression ratios since most of the signal energy is concentrated in a limited number of significant coefficients" (Ciocoiu, 2009).

3.2.4 ECG and Cardiospy

Cardiospy has the modules which analyze the ECG signals recorded by the Holter system of Labtech Ltd. So, it can mark the annotation of beats, namely it can perform the detection of QRS complexes.

Cardiospy can classify heartbeats. In the following section only 3 categories are mentioned, the normal heartbeats denoted by N, the ventricular beats denoted by V, and the supraventricular beats denoted by S.

3.2.5 Clustering and visualization of ECG module of Cardiospy

The goal of the new module of Cardiospy is to cluster and visualize the long (up to 24-hours) recordings of ECG signals, because the manual evaluation of long recordings is a lengthy and tedious task. The module put seemingly similar heartbeats into one group. Thus, cardiologists do not have to examine all the (often more than 100000) heartbeat curves to find the heartbeats which morphologically differ from the normal beats. They only have to analyze groups belonging to abnormal beats. On the one hand, the task of the cardiologists is becoming simpler; on the other hand, the possibility of making a mistake is reduced as they discover abnormal beats more easily. In the module, there are automatic and manual clustering features. (Vágner et al., 2011 A), (Vágner et al., 2011 B), (Juhász et al., 2009)

I worked together on this project with László Farkas (Labtech Ltd.), István Juhász (University of Debrecen, Faculty of Informatics), and two students from the Faculty of Informatics, University of Debrecen, József Kuk and Ádám Balázs.

We developed the clustering program in C# 4.0 programming language, in Visual Studio 2010 environment.

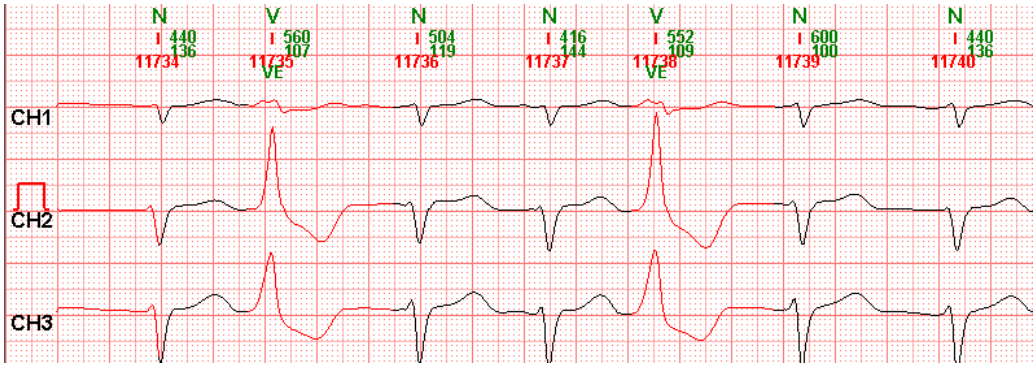

Figure 26: Three channel ECG recording

3.2.5.1 Input data

Figure 26 shows a part of a three channel ECG recording. Normal heartbeats are marked with 'N', ventricular heartbeats with 'V'. On the figure you can see a vertical line at every heartbeat marking the annotation of the beat.

3.2.5.2 Processing

The program processes the digitalized, raw ECG signals with a methodology that is similar to the methodology discussed in (Micó et al., 2005). The difference between the two methodologies is that we apply wavelet transformation instead of polygonal approximation.

In the first step, the QRS detector locates the specific position of the heartbeats. The annotation of a heartbeat helps to find the whole ECG signal part which belongs to a heartbeat, so we can split the ECG signal into signals each of which belongs to a heartbeat. But this signal has a lot of points (if the sampling rate is 256, a heartbeat has about 256 points), which are not easy to be clustered. Moreover, the signal has noise, so we have to use a filtering technique.

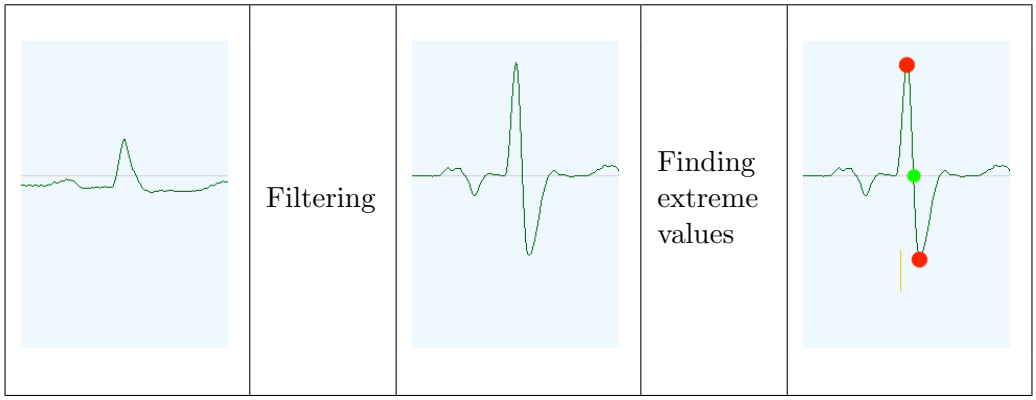

Figure 27: Representing an ECG signal with characteristic points

Our idea is that a heartbeat can be characterized with only a few points. In our first solution, we differentiate the ECG signals as a filtering, then we find the reference point. The reference point is the first intersection of the x axis and the signal, after the annotation. It is the green point on Figure 27. Next, we find the extreme values after and before the reference point. The red points represent the extreme values and its places. We transform the three points into a coordinate system in such a way, that the green point is at the origin. So, the two red points represent the filtered signal. In this way we have information about the

width and height of the filtered signal. As a result, we can work with a pair of two-dimensional points instead of the signal. (Vágner et al., 2011 B)

On the other hand, we find that this solution deals the noise not well. The wavelet transformation (Ciocoiu, 2009) is a better solution for filtering and it helps in the identification of the specific points which characterize the ECG waveforms well. As a result, we also characterize a heartbeat with a pair of two-dimensional points.

Figure 28 shows the point pairs of a recording in a coordinate system. The figure shows that the points create arbitrary shaped clusters.

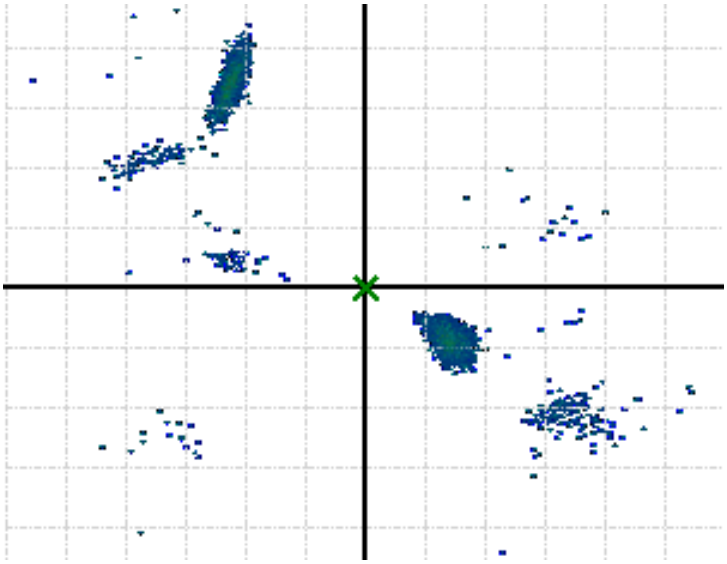

Figure 28: The set of characteristic points of ECG signals of a recording

We apply an early, special version of the GridOPTICS algorithms. In our case many points in the set of points have the same coordinate or they are close to each other. With the grid-based method we can radically reduce the number of points and the runtime of our algorithm, namely the user does not need to wait for the result of the clustering.

The main reason that many points have the same coordinate is that the dispersion of points representing different types of heartbeats is not entirely random. They create well-separable sets of points where the sets of points are very dense. There are only a few stand-alone points that are not clusterable.

As a result of the algorithm, clusters of heartbeats appear. We put the not clustered points into a special garbage cluster.

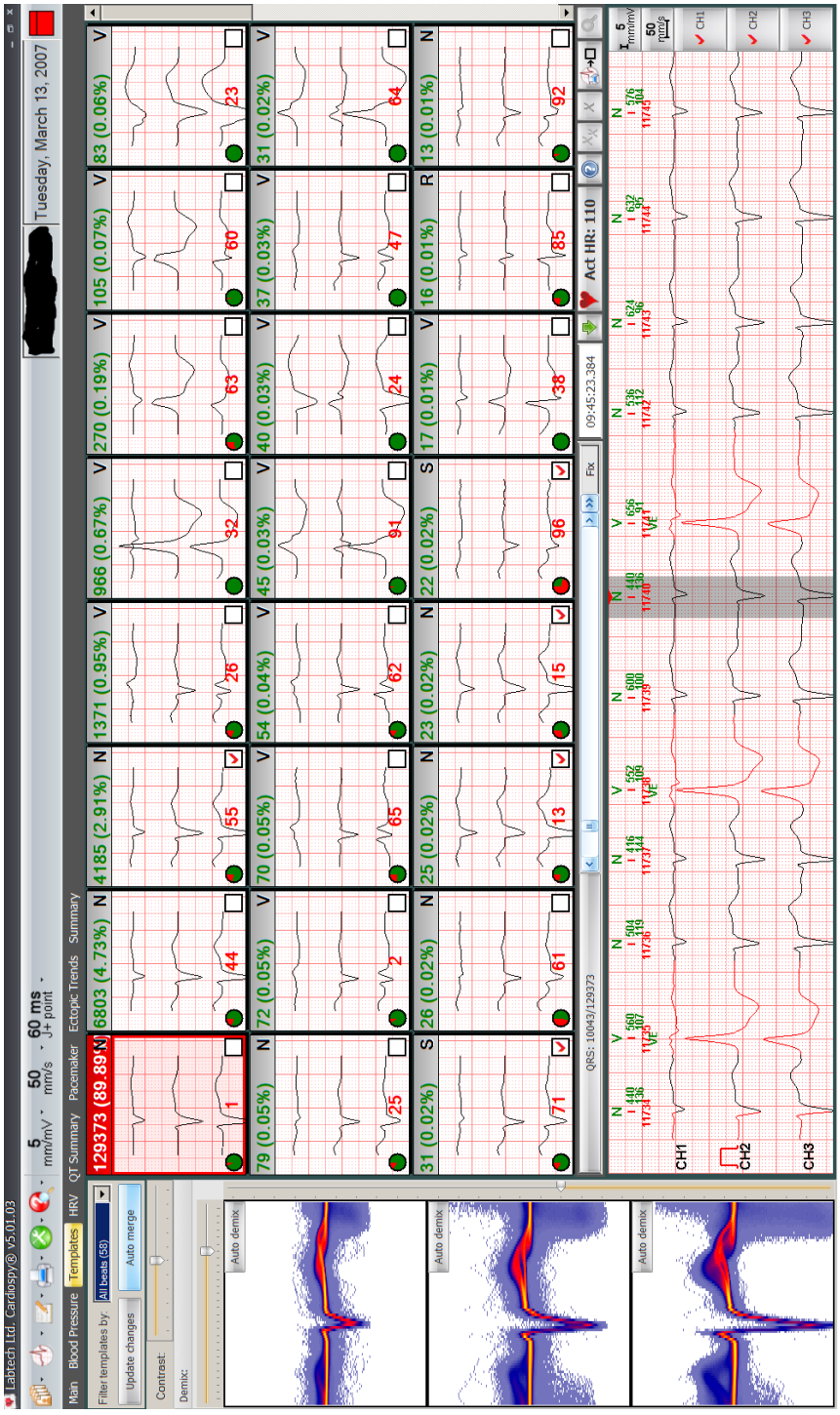

Figure 29: Full screen of the module of Cardiospy which visualizes and clusters of ECG signals

We characterize every cluster by the median of curves. The median of curves is produced a way that for every x coordinate in the interval of the curves of a cluster the median point is chosen from the points which has the x coordinate and which belongs to one of the curves. We call the median of curves template. We can analyze curves belonging to certain clusters together and separately too by using visualization devices. Heartbeats belonging to certain clusters can be examined one by one in their original environment.

3.2.5.3 Visualization

The program provides an interactive graphical user interface in which the results of clustering are visualized. So, the cardiologist can analyze them, manage them, and can work on them further.

In Figure 29, you can see the full screen of the program. In the right upper part we visualize the templates of certain heartbeat clusters. On the left side you can see the heartbeats drawn on each other belonging to the template marked with red color. In the right down part you can see the heartbeats belonging to the template and their exact position on the ECG recording.

We can visualize the main features of certain heartbeat groups by the help of the templates. The templates can help to look through all the heartbeat groups. As a main feature, the left upper corner of the template appears how many heartbeats are there in the template and beside it the percentage compared to the total heartbeat number. In the upper right corner you can see the type of the heartbeats. In the left lower corner the pie chart shows how the heartbeats in the group are similar to each other. The more green color it contains, the more resembling the heartbeats are. The red number in the center shows the identifier of the heartbeat group. The square in the right lower part helps the cardiologists. They can put a check mark in it if they have already analyzed that group.

In the right down part of Figure 29, you can see the whole recording in an enlarged form. You can go through the elements of the cluster marked by red color with the help of the scroll bar in the upper part. The heartbeat of the grey column is an element of the heartbeat cluster marked by red color.

On the left side of Figure 29, we draw heartbeats on each other belonging to the cluster denoted by red on the right part of the window. The starting points of the drawings on each other are the annotations. We can grab every heartbeat at its annotation. After this we cut down areas with a given interval from the left and right side of the recording and we represent certain curves in this way. If there are more heartbeats at the given area, its color first becomes darker and after that

its color becomes redder and redder. The aim is that the more heartbeats go to an area, the more powerful the representation should be.

To make its use easier, the figure gets 3 sliders. You can alter contrast with the upper horizontal slider. You can shrink and stretch with the lower horizontal slider in horizontal direction, whereas with the vertical slider in vertical direction. Vertical and horizontal sliders constitute a great help in manual clustering.

3.2.5.4 Manual clustering

You can divide certain clusters into further groups manually. You can select the heartbeats in order to be cut by the mouse. By the effect of these, the program divides the cluster into two groups. The selected heartbeats constitute a cluster; all other heartbeats constitute another one. In the left part of Figure 30, you can see the heartbeats drawn on each other and templates belonging to the original cluster. In the right part of Figure 30, you can see the curves drawn on each other and the templates belonging to the two new clusters. We performed the manual clustering on heartbeats of the first channel.

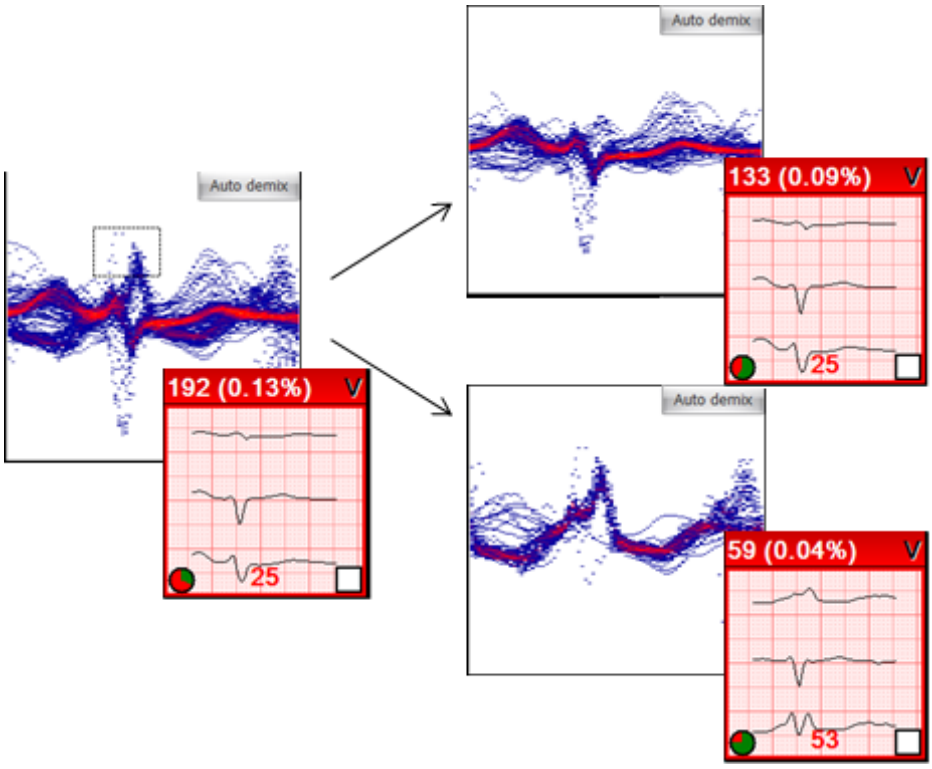

Figure 30: Manual clustering in Cardiospy

3.2.5.5 Working with the recordings

After the physician has loaded a recording, Cardiospy clusters its heartbeats using its first channel. The clustering algorithm puts the dominant heartbeats into a few clusters and it may put the abnormal heartbeats into a lot of clusters, because they are hardly similar to each other and to the dominant heartbeats.

In the recordings there can be heartbeats which are not easy to classify as V – ventricular, S – supraventricular or noise. The clustering algorithm cannot decide the classification of the heartbeats, but it puts the similar abnormal heartbeats into a cluster. If the physician thinks that the type of a heartbeat or the heartbeats of the cluster decided by Cardiospy is wrong, they can change its type.

The physician can work on with the clusters. They can analyze the heartbeats of each cluster visually together or they can consider each heartbeat of a cluster in its original place on the ECG. Moreover, the physician can split the clusters further manually or with the clustering algorithm using the same or another channel.

3.3 Blood pressure measurement

The blood pressure is the amount of blood pumped by the heart in relation to the size and condition of the arteries. The blood pressure measurement is the measure of the force of blood on artery walls, which is given in millimeters of mercury (mmHg). Blood pressure generally refers to arterial pressure.

The blood pressure measurement determines two values: the *systolic pressure* (SP) and the *diastolic pressure* (DP). The blood pressure is the highest during the systole, when ventricles contract, whereas the blood pressure is the lowest during the diastole, when ventricles relax and refill.

The *cardiac cycle* is a complete cycle of heart events, it begins of the first heartbeat and ends with the beginning of the next heartbeat. The systolic pressure is the highest recorded arterial pressure, which occurs near the beginning of the cardiac cycle, whereas the diastolic pressure is the lowest recorded arterial pressure, which occurs in the resting phase of a cardiac cycle.

The cardiac cycle can be broken down into four phases. The first phase is the atrial systole, which occurs when the atria are electrically stimulated and is denoted as the P-wave in an ECG. This stimulation causes the atria to contract. The second phase is the ventricular systole, which occurs when the ventricles are electrically stimulated and is denoted as the QRS-wave segment in an ECG reading. This stimulation causes the contraction of the ventricles, so here we can read the systolic pressure. The third phase is the early diastole. It is when the heart begins to relax after its stimulation and is denoted as the T-wave in an ECG. Here the ventricles relax. The fourth phase is the diastole, when the heart finishes up its relaxation period; this moment is denoted as the TP-period in an ECG. We can read the diastolic pressure from the diastolic period of the phases of the cardiac cycle.

The *mean arterial pressure* (MAP) is the average pressure throughout a cardiac cycle, whereas the *pulse pressure* is the difference between the maximum and the minimum pressures.

The *normal blood pressure* values are when the systolic pressure is under 120 mmHg and the diastolic pressure is under 80 mmHg. However, if the systolic pressure is under 90 mmHg or the diastolic pressure is under 60 mmHg, it is called *hypotension*, which needs medical examination. If the systolic pressure is between 120 and 139 or the diastolic pressure is between 80 and 90, and these values are measured more times, it is called *pre-hypertension*. It does not need any treatments, but the patient has a high risk of developing hypertension. If

the systolic pressure is more than 139 or the diastolic pressure is more than 90, and these values are measured several times, it is called *hypertension*. It needs medical treatment, because it can cause several illnesses or it can be related to ones, such as heart attacks, heart failure, aneurysms of the arteries, stroke, brain damage, blindness, renal failure, peripheral arterial disease, chronic kidney disease, hormonal changes, endocrine problems, etc. (Welch Allyn, 2015)

3.3.1 Types of blood pressure measurements

3.3.1.1 Invasive method

The invasive blood pressure measurements mean direct measurements of the arterial pressure, so a cannula needle is placed into an artery. These methods are generally used in intensive care units of hospitals. On the other hand, they are known to carry a small risk and they can be used only under sterile circumstances. They are used, because they have a high degree of accuracy and they provide information during an operation continually, namely beat-to-beat. (Sorvoja, 2006), (Welch Allyn, 2015).

3.3.1.2 Noninvasive methods

Noninvasive blood pressure measurement devices are increasingly becoming popular both in clinics and in home care because of their affordable price and easy usage due to their automatic measurement features. Nevertheless, the accuracy of these devices has not yet reached the necessary level. (Sorvoja, 2006)

There are a lot of types of noninvasive blood pressure measurement methods, but two of them are well-known: the auscultatory and the oscillometric methods.

3.3.1.3 The auscultatory method

Korotkoff used the auscultatory method for the first time. Physicians need a mercury manometer and a stethoscope. First they wrap a cuff around the upper arm, then they quickly raise cuff pressure until it stops blood circulation on the distal side of the hand, indicated by palpating the radial artery. During the following slow pressure drop, audible sounds can be heard through the stethoscope, which was placed on the skin beyond the brachial artery. These sounds were affected by the blood wave in the artery under the cuff, and were audible at 10-12 mmHg, slightly before the pulse can be palpated on the radial artery. At this point, cuff pressure is taken to indicate maximum blood pressure, while minimum

blood pressure is achieved when the murmur sounds disappear. The sounds name is Korotkoff sounds. (Geddes 1991)

The appearance and disappearance of sound can be used to determine systolic and diastolic blood pressure, respectively, while the cuff deflates. However, establishing the point at which sounds disappear is not always obvious; in fact, misleading readings are easy to record. Thus, a certain sound intensity level is often used to determine the point corresponding to diastolic blood pressure. Also the sound amplitude maximum can be used to determine mean arterial blood pressure. (Sorvoja, 2006)

This technique measures the systolic and the diastolic pressure. The mean arterial pressure is an estimated value. (Welch Allyn, 2015)

Measurements based on the auscultatory method are difficult to automate, because the frequency spectrum of the different phases of Korotkoff sounds is closely related to blood pressure. When a patient's blood pressure is high, also the recorded frequency spectrum is higher than normal and decreases as a function of blood pressure. With hypotensive patients and infants, on the other hand, the highest spectrum components can be as low as 8 Hz, which is below the human hearing bandwidth. (Sorvoja, 2006)

In ambulatory measurements, when the patient is able to move moderately, noise may become dominant, thereby spoiling the measurement. This may be avoided by using two identical microphones under the cuff, one located on the upper side, the other on the distal side. Ambient noise reaches both microphones at the same time, but the blood pressure pulse propagating through the brachial artery arrives after a time delay. This phenomenon can be used for noise cancellation. (Sorvoja, 2006)

3.3.1.4 The oscillometric method

In the case of oscillometric method a cuff is also wrapped around the upper arm similarly to in the case of the auscultatory method, but there is a sensor built in the cuff which perceives the pressure of the artery and the cuff. The cuff pressure is raised until it stops blood circulation on the distal side of the hand, indicated by palpating the radial artery. Then the air from a cuff gradually deflated (continuously or by steps). During this process, weak pulsations (up to 5 mmHg) of the pressure occur in a cuff due to pulsing of blood pressure in the artery, flowing under a cuff. These small measurements, termed "oscillometric pulses", are recorded in all the pressure range in a cuff. (BPLab, 2015)

If there are steps in the deflation of the air, they increase noise immunity, because there is a possibility "to wait" at the next step of air deflation, until the interference or episode of an arrhythmia disappear. (BPLab, 2015)

After removing the pressure values caused by the deflation of the cuff there will be an oscillation waveform called oscillogram. The pressure in a cuff is drawn on the horizontal axis, and relevant values of amplitudes of pulses are drawn on the vertical axis. First the changes of the pressure on the waveform increase, then decrease. A curve can be fitted into the minimum and the maximum pressure values, which will form a "bell". The shape of the "bell", is varying from patient to patient (and sometimes it varies for one patient from minute to minute). Under correct measurement conditions, "the bell" has a single, legibly expressed maximum. (Ball-llovera et al., 2003), (Wang et al., 2002), (Lin, 2007), (Sorvoja, 2006)

Mean arterial pressure (MAP) is defined as the pressure in a cuff, at which the maximum amplitude of "oscillometric pulse" (i.e., according to the position of the maximum amplitude of "the bell") was detected (Sorvoja, 2006). The oscillometric method calculates the value of the systole, the diastole based on the changes of the pressure on the oscillogram (Ball-llovera et al., 2003) (Wang et al., 2002) (Lin, 2007). So, the oscillometric method detects the mean arterial pressure directly (unlike auscultatory method), but the systolic and diastolic pressures are calculated. (Welch Allyn, 2015)

Figure 31 shows an example of the cuff pressure curve and the corresponding oscillometric waveform.

The values of blood pressure can be determined based on the cuff press curve at a given point. The method determines the point of the systole, the diastole, and the mean arterial pressure. The point of the mean arterial pressure is the maximum point of the oscillogram. There are two algorithms to determine the point of the systole and the diastole: the height-based method and the slope-based.

The slope-based method fits a curve to the changes of cuff pressure. The method specifies the inflection point of the curve as the points of systole and diastole (Ball-llovera et al., 2003), (Sapinski, 1997), (Lin, 2007).

The height-based algorithm has two previously given ratios, one for the systole and the other for the diastole. The two ratios are not necessarily the same. The pressure change at the point of the mean arterial pressure is 100%. The method finds the point where the pressure change corresponds to the given ratio. The point of systole is before the mean arterial pressure, whereas the diastole is after it (Ball-llovera et al., 2003), (Lin et al., 2003), (Lee et al., 2001) (Lin, 2007).

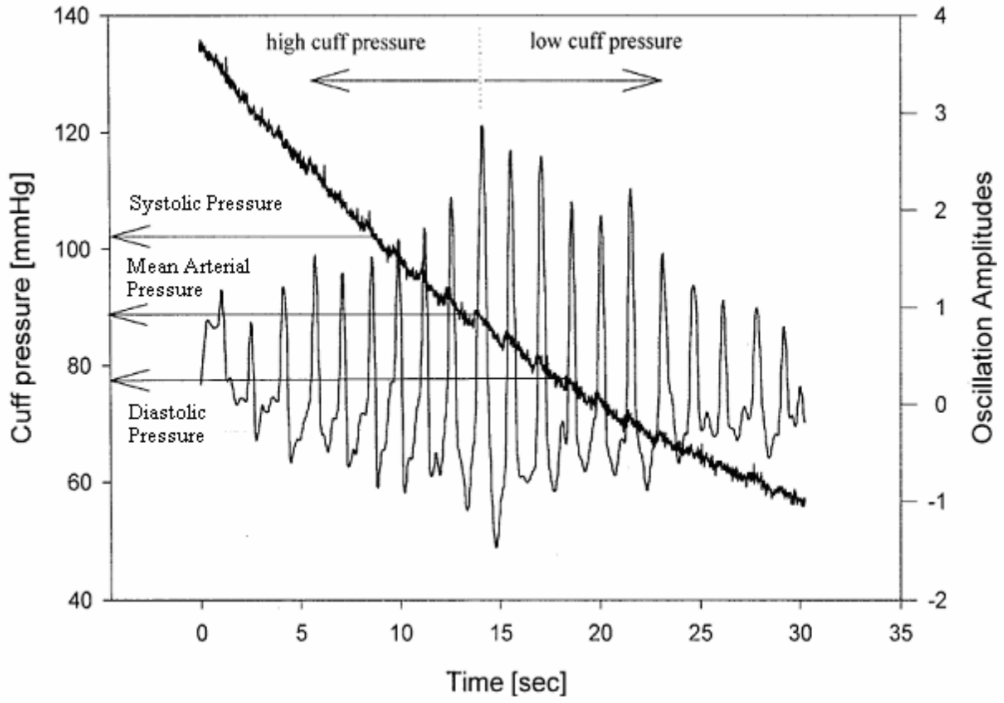

Figure 31: Cuff pressure signal and oscillation waveform (Lin et al., 2003), (Lin, 2007)

The values of the ratio are not exactly determined. It depends on the realization of the oscillometric algorithm. Most researchers put these values between 40% and 60% (Ball-llovera et al., 2003), (Lin, 2007), (Sapinski, 1992), (Geddes, 1991). (Sorvoja, 2006) states that the systolic ratio should be lower for hypertensive patients and that the diastolic ratio should be lower for hypotensive patients. He also claims that a measurement device with fixed ratios for determining systolic, mean and diastolic blood pressure may significantly overestimate these values.

The oscillometric method is realized in many ways, like prediction and smoothing algorithm, fuzzy logic, neural network, pattern recognition, mathematical modelling, etc. (Lin, 2007).

Most automatic blood pressure measurement devices implement the oscillometric method. These devices are cheap, so most people can buy them. They are usually used at home, but they are also popular in hospitals when only a simple control is needed.

3.3.2 The causes of errors in oscillometric blood pressure measurements

There are several problems which can cause noise signals or incorrect results. They are:

1. Incorrect position of the cuff relative to heart. When measuring blood pressure the middle of the cuff that is placed on the patient's upper arm should be at heart level. When the middle part of the cuff is displaced from the heart level, an error occurs. When the cuff is displaced downwards relative to the heart level, the measured blood pressure is increased and vice versa.
2. Incorrectly selected cuff. The pneumo camera of the cuff should cover not less than 40% of an upper arm circle and not less than 80% of its length. Using a narrow or short cuff results in essential false higher value of blood pressure.
3. Not tightly applied cuff. According to the rules of blood pressure measurement there should be a finger-wide gap between a cuff and the surface of the upper arm of the patient. Untight application of the cuff results in some negative effects, they can be:
   - The contact area of pumped cuff with the surface of an upper arm is decreased, which gives the same effects as in the case of using narrow cuff.
   - The inflation rate is decreased, which results in violation venous blood outflow and causes pain sensations.
   - The power consumption by the monitor pump is increased, which can cause the discharge of the batteries before the end of the monitoring.
4. Expressed violation of a rhythm. When a rhythm is disturbed the filling of vessels by blood becomes irregular. When using the oscillometric method the shape of "the bell" is distorted, and the error of the measurement of all blood pressure parameters is increased. The mathematical methods of data smoothing, which are used in modern automatic blood pressure meters, are effective in this case.
5. Blood pressure measurement for elder people. Thickening and inspissation of the wall of a humeral artery is observed on elder people, which means that humeral artery becomes rigid. Higher pressure level is required in the cuff (higher than the arterial pressure) for reaching a compression of rigid arteria. As a result there is a false overestimation of the blood pressure level (phenomenon "pseudo-hypertonia"). Palpation of the pulse on radial artery at a level of pressure in cuff, exceeding a systolic blood pressure, helps to

recognize this error. It is necessary to define blood pressure on a forearm by palpation.

6. Very large circle of an upper arm (obesity, very developed musculature). The precision measurement of blood pressure may be impossible for the patients with a circle of an upper arm more than 41 cm or with a conical shape of the upper arm, which means that it is impossible to apply the cuff correctly.
7. Arm movement. If the patient moves his arm, exerts or relaxes it during measurement, it causes changes of pressure in the cuff, connected with a systole. In the presence of such changes, the instrument cannot correctly recognize pulses.
8. Walking during measurement. Pressure pulsations in the cuff, which occur during walking, are superimposed on pulses connected with a systole. These are two different sequences of pulsations, having different origin, and it is difficult to separate them because their amplitudes and period are very similar. As a result, similarly to the case of rhythm disturbance, the measurement time is increased; the shape of "the bell" is distorted, which causes considerable blood pressure measurement errors.
9. The respiratory waves are periodic changes of blood filling of the vessels synchronous to respiration. They more often happen in corpulent patients. During blood pressure measurement the dependence of pulsations amplitude from the pressure in a cuff is distorted by changes of the amplitude connected with respiratory waves. It produces distortions of the shape of "the bell" of pulses amplitude in the form of valleys or plateau.
10. The plateau on the "bell". In some cases the top of the "bell" of pulses amplitudes has the shape of a plateau, i.e. it has no expressed maximum. It causes an error of determining mean arterial pressure, and consequently, systolic and diastolic pressure. Plateau can be caused by the episodes of arrhythmias and respiratory waves. Another cause can be the rigidity of arteries, which causes the effects connected with reflection of pulse wave in partly over-compressed vessels. (BPLab, 2015)

3.3.3 Oscillometric technique of Aboy (2011)

Aboy (2011) obtained a patent which includes a method for blood pressure measurement from noninvasive oscillometric pressure signals. Figure 32 shows a diagram of his method and Figure 33 shows the curves of each step.

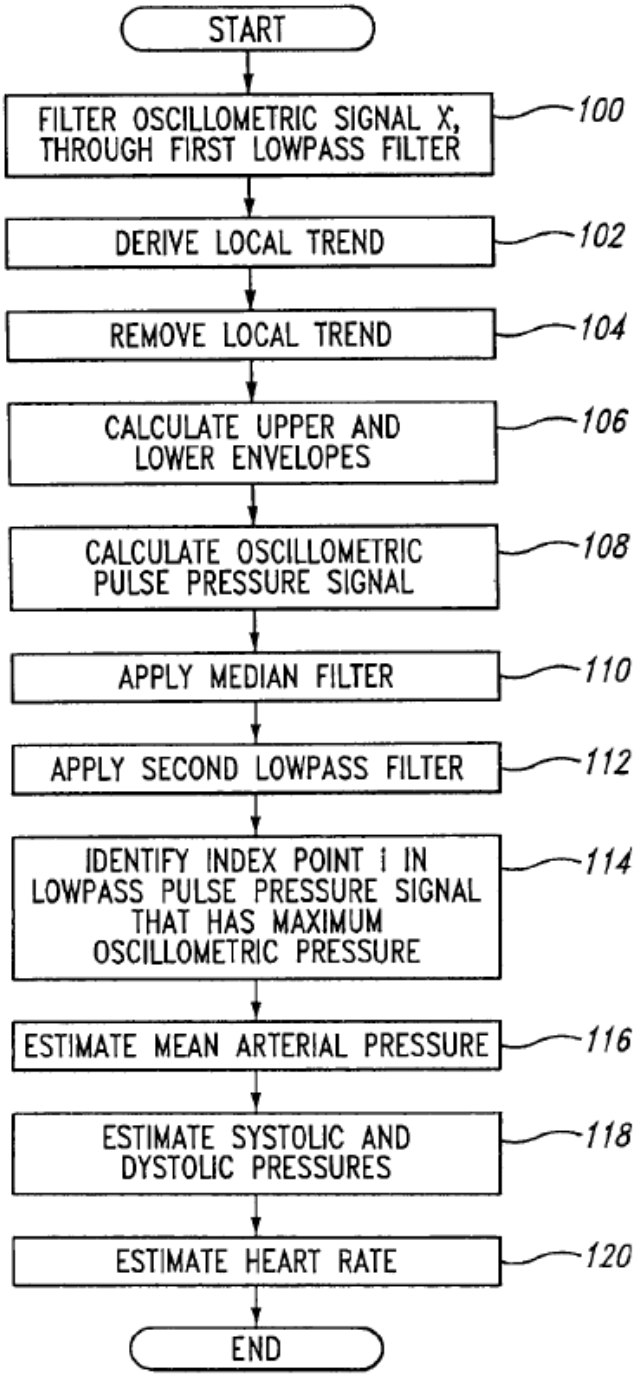

Figure 32: The main steps of the oscillometric technique of Aboy (2011) (Original picture)

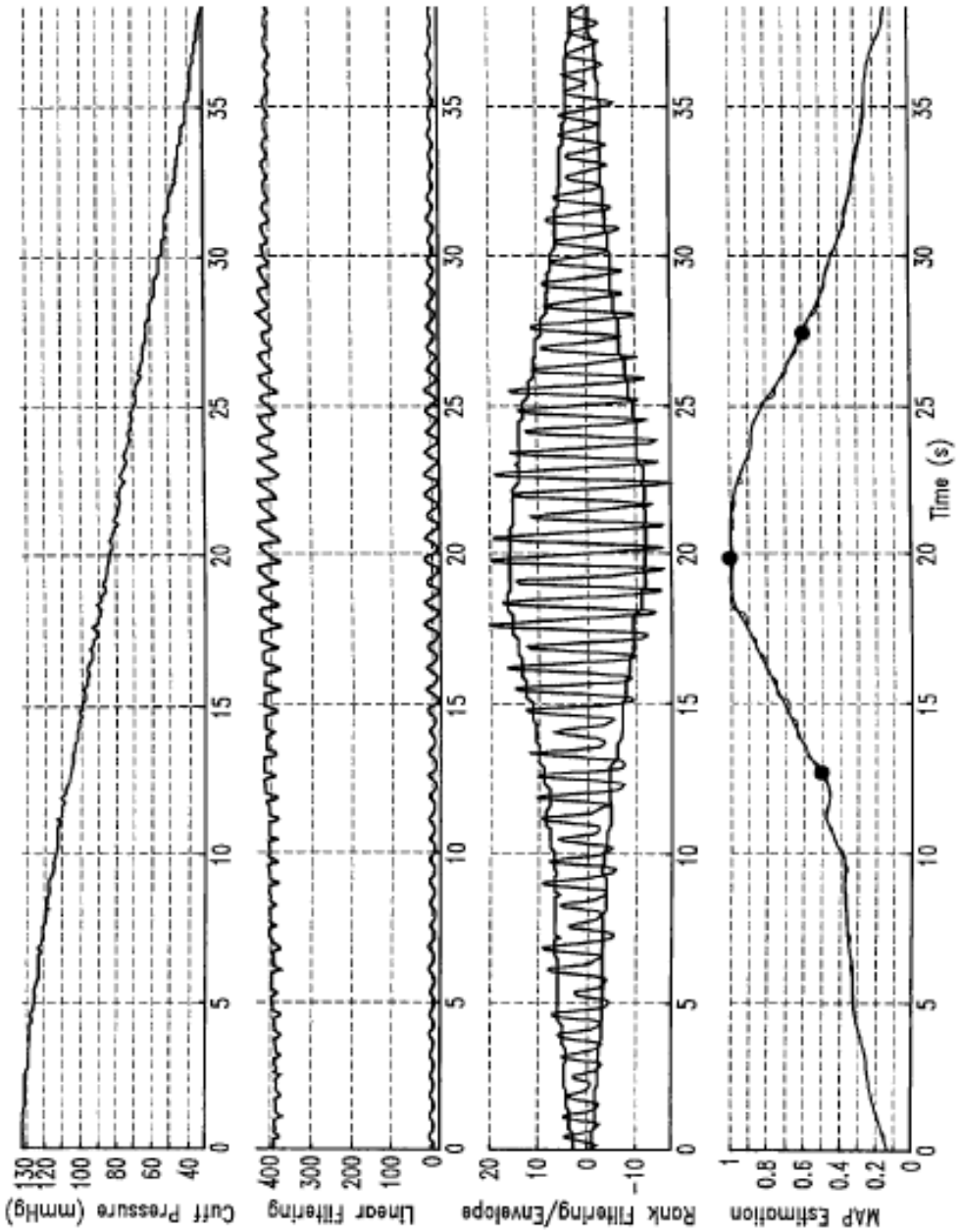

Figure 33: The curves belong to the main steps of the oscillometric technique of Aboy (2011) (Original picture)

The method uses an oscillometric signal and a cuff pressure signal as input. The upper part of Figure 33 shows the cuff pressure signal. In step 100 a lowpass

filter is used on the input oscillometric signal to remove high frequency noise and artifact. In step 102 the local trends of the lowpass filtered oscillometric signal are estimated using a linear filter. In Step 104 local trends are removed from the filtered oscillometric signal. The result curves of these steps are displayed on the second part of Figure 33. In step 106 the upper and lower envelopes of the detrended and lowpass filtered oscillometric signal are estimated using a rank-order filter. The third part of Figure 33 shows the two envelopes.

In step 108 the oscillometric pulse pressure signal is calculated, namely the lower envelope is subtracted from the upper envelope signal. As a result we get the pulse pressure signals on the fourth part of Figure 33. In step 110 a median filter is applied to the pulse pressure signal to remove components due to artifact. In step 112 a lowpass filter is applied on the pulse pressure signal to remove high frequency components due to artefact.

In step 114 the location of the maximum oscillometric pulse pressure is identified. In step 116 the mean arterial pressure is estimated by finding the cuff pressure at the location of the maximum oscillometric pulse. In step 118 the systolic and diastolic pressures are estimated by identifying the two (systolic and diastolic) percent points preceding and following the location of the maximum oscillometric pulse on the pulse pressure signal and identifying them in the cuff pressure signal. The locations of the maximum oscillometric value, and the systolic and diastolic values are denoted by black points on the pulse pressure filter on the fourth part of Figure 33.

In step 120 the heart rate is estimated by finding the frequency corresponding to the maximum spectrum amplitude in the range of physiological interest.

3.3.4 PC-side oscillometric blood pressure measurement

The blood pressure values which are measured by a microcontroller equipped with an oscillometric blood pressure measurement can be inaccurate and it does not inform the patient and the doctor appropriately. However, this method needs only simple devices and can be used in an automated way.

Moreover, there is a demand for observation of blood pressure during a day, which means that the device measures the blood pressure every 15-20 minutes. The physicians can discover more information about the status of the patient. Ambulatory blood pressure measurements provide important prognostic information about cardiovascular risk, mortality and progression of hypertensive

end organ damage and is a better predictor of risk compared to office blood pressures. (Ghuman et al., 2009)

There is also a demand for blood pressure measurement during sport activities and in clinical use.

As it is introduced above, there can be a lot of types of errors during the oscillometric method, which cannot be recognized by a microcontroller. Furthermore, the evaluations of the long-term (24 hours long) recordings are also difficult with a microprocessor.

Sending the recordings collected by a microcontroller to an application which runs on a PC can be a good solution. A recording can contain only one measurement or a sequence of measurements created during 24 hours. The advantage of the PC side application is that it can use more memory and processor capacity, so it is faster and more precise.

With the help of the application the physicians can recognize the types of errors on the measurements, they can find the noisy measurements, and the application can offer some other features for its users.

3.3.5 BP Service module of Cardiospy

The goal of the new BP Service module of Cardiospy is to calculate and visualize the values of blood pressure. It visualizes the result of the steps of an oscillometric algorithm, then it determines the values of the blood pressure based on the method of Aboy (2011). The algorithm decides whether the result is acceptable and authentic based on the characteristic of the recording. The module can process the long-term recordings.

The BP Service module needs to be validated with the help of an application. It executes the algorithm on mass of the recordings which have reference measurement values. The application shows the differences between the results of the algorithm and the reference values. The application helps to qualify the algorithm according to the international standards. (Vágner et al., 2014)

I worked together on this project with Péter Tóth (Labtech Ltd.), István Juhász (University of Debrecen, Faculty of Informatics), and two students of the Faculty of Informatics, University of Debrecen, Béla Vámosi and Dávid Angyal.

We developed the BP Service module and the validation application in C# 4.0 programming language, in Visual Studio 2010 environment.

3.3.5.1 The input data

The input data is an oscillometric recording which is created by a microcontroller during blood pressure measurements. It is a C8051F064 mixed-signal MCUs of Silicon Laboratories. It is a reliable high-speed 8051 architecture MCU with two 16-bit ADCs. It has 64kB Flash memory which is in-system programmable in 1kB sectors and it has 4kB data RAM. The recorded data is stored on the micro SD^{TM} card. (Silabs, 2015)

During a measurement the deflation of the cuff pressure is continuous. A recording can contain only a measurement or a sequence of measurements created during 24 hours.

The beginning and the end of each measurement, the beginning of the deflation of each measurement and the sample rate are given in the input files. The sample rate of most input recordings have 200 Hz.

3.3.5.2 My oscillometric blood pressure measurement algorithm

I based my method on the oscillometric method of Aboy (2011). The main difference between the method of Aboy (2011) and my method is that I fit a polynomial instead of two wrapping curves. The idea is given by the article of Zheng et al. (2011), who determine the mean arterial pressure from the peak of the 6th order polynomial model envelope fitted to the sequence of oscillometric pulse amplitudes.

The main steps of the algorithm are:

1. The algorithm splits the recording into measurements. It processes only one measurement at a time. The algorithm processes the part of the measurement after the beginning of the deflation.
2. The algorithm creates an oscillogram based on the measurement using a band-pass filter.
3. The algorithm finds the local minimum and maximum points and values.
4. It creates a histogram from the local extrema. The histogram shows the change of the cuff pressure at a minimum point.
5. The algorithm fits a wrapping curve which is a polynomial to the histogram.
6. It determines the maximum point and its place of the wrapping curve.
7. It determines the systolic pressure and the diastolic pressure both height-based and slope-based.
8. Based on the character of the measurement it gives information whether the result is acceptable or not. In the result of the algorithm there is a sign

which shows that the blood pressure values are acceptable, not acceptable, or they can be accepted only after manual analyzing.

3.3.5.2.1 Band-pass filter

There are two main goals of the digital filters. One is to remove the noise from the signal. The other one is to separate the signal into two or more signals. In our case both of them are useful. Firstly, in the input signals there are the pressure values which are caused by the deflation of the cuff and there is the oscillogram which represents the pressure of the artery. Secondly, the signals can also be noisy.

The low-pass filter is a filter which passes signals whose frequency is lower than a cutoff frequency and attenuates signals whose frequency is higher than the cutoff frequency. The low pass filter can be calculated in the following way:

$$y_1 = x_1, \quad y_i = \alpha x_i + (1 - \alpha) y_{i-1},$$

where y_i, $i = 1 \dots n$ are the filtered values, x_i, $i = 1 \dots n$ are the original values. α is the parameter of the filter, and $0 \leq \alpha \leq 1$.

The high-pass filter is a filter which passes signals whose frequency is higher than a cutoff frequency and attenuates signals whose frequency is lower than the cutoff frequency. The high pass filter can be calculated in the following way:

$$y_1 = x_1, \quad y_i = (1 - \alpha)(y_{i-1} + x_i - x_{i-1}),$$

where y_i, $i = 1 \dots n$ are the filtered values, x_i, $i = 1 \dots n$ are the original points. α is a parameter of the filter, $0 \leq \alpha \leq 1$.

The band-pass filter is a combination of low-pass filter and a high-pass filter.

We determined the values of α in both high-pass and low-pass filter approximately. We had to pay attention to the fact that maximum and minimum peaks of the filtered signals have to be in the same places as they were in the original signals.

3.3.5.2.2 Find local maximum and minimum values

In order to determine whether a point is a local maximum or minimum, we need an ϵ-neighborhood of the point. The main question is what the value of ϵ is. First of all, it depends on the sample rate. Then if the patient has a high pulse, it cannot be too large. But if it is too small, the algorithm will consider a lot of noisy points as local extrema.

3.3.5.2.3 The wrapping curve

The wrapping curve is a 6th order polynomial. So, the model of the wrapping curve is:

$$y = \sum_{i=0}^{6} a_i h_i(x), \quad h_i(x) = x^i$$

or:

$$y = \mathbf{h}(x)\mathbf{a}, \quad \mathbf{a} = (a_0, \ldots, a_6)^T, \quad \mathbf{h}(x) = (h_0(x), \ldots, h_6(x)).$$

We use the method of the least squares, which means that the following expressions should be minimal:

$$F = \sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{6} a_i x_j^i \right)^2$$

or

$$F = \sum_{j=1}^{n} \left(y_j - \mathbf{h}(x_j)\mathbf{a} \right)^2,$$

where (x_j, y_j), $j = 1 \ldots n$ are the input points.

In order to find a_6, a_5, ..., a_0 values of the wrapping curve, we have to differentiate F. The values for which the first derivative of F are equal with 0 can be local extremum (in this case minimum). In this way we get the following normal equation, whose solutions are the a_6, a_5, ..., a_0 parameter values of the wrapping curve.

$$(\mathbf{H}^T\mathbf{H})\mathbf{a} = \mathbf{H}^T\mathbf{y}, \quad \mathbf{y} = (y_1, \ldots, y_n)^T, \quad \mathbf{H} = (\mathbf{h}(x_1), \ldots, \mathbf{h}(x_n))^T$$

So, it the matrix $\mathbf{H}^T\mathbf{H}$ is non-singular, we have

$$\mathbf{a} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}.$$

(Bishop, 2006)

3.3.5.2.4 Determination the maximum point of the wrapping curve and the pressure values with the slope-based method

We differentiate the wrapping curve. There will be the place of the maximum where the differentiated curve is 0.

There will be the place of the inflections points (and the place of the systolic and diastolic pressure) where the second differentiated curve is 0 and which are not the maximum values.

In both cases these conditions are only necessary, but not sufficient, however, the curve is a bell, so it does not need further examination. If the algorithm gives wrong results or errors the visualization interface shows it immediately.

If the place of the systolic and diastolic pressure is given, the algorithm considers the values of the original cuff pressure curve in the given places as systolic and diastolic values.

3.3.5.2.5 Determination the maximum point of the wrapping curve and the pressure values with the height-based method

There are two ratios for both systolic and diastolic values. The algorithm considers the place of the bar of the histogram which value is the highest of the values which are less than the maximum value multiplied by the ratio as the place of the systole. Similarly it considers the place of the diastolic pressure with the other ratio.

The algorithm calculates the pressure values as in the previous case.

3.3.5.2.6 BP Service visualization interface

The BP Service realizes a unified, structured, and interactive data visualization. The user can turn various data elements on and off using controls. If a control is set the surface is drawn again in an interactive way.

The BP Service uses the results and the data of other modules of the Cardiospy system (e.g.: ECG, auscultation blood pressure). On the surface they appear, but they are not relevant to the topic of the article.

We use colors on the surface to distinguish the curves, the lines, and other information. For example the curve of the cuff press is green, the oscillogram is yellow.

The visualization surface can be divided into 4 main parts based on functions see Figure 34. These are:

1. the panel for displaying the results,
2. the canvas,
3. the buttons,
4. the oscillometric control page.

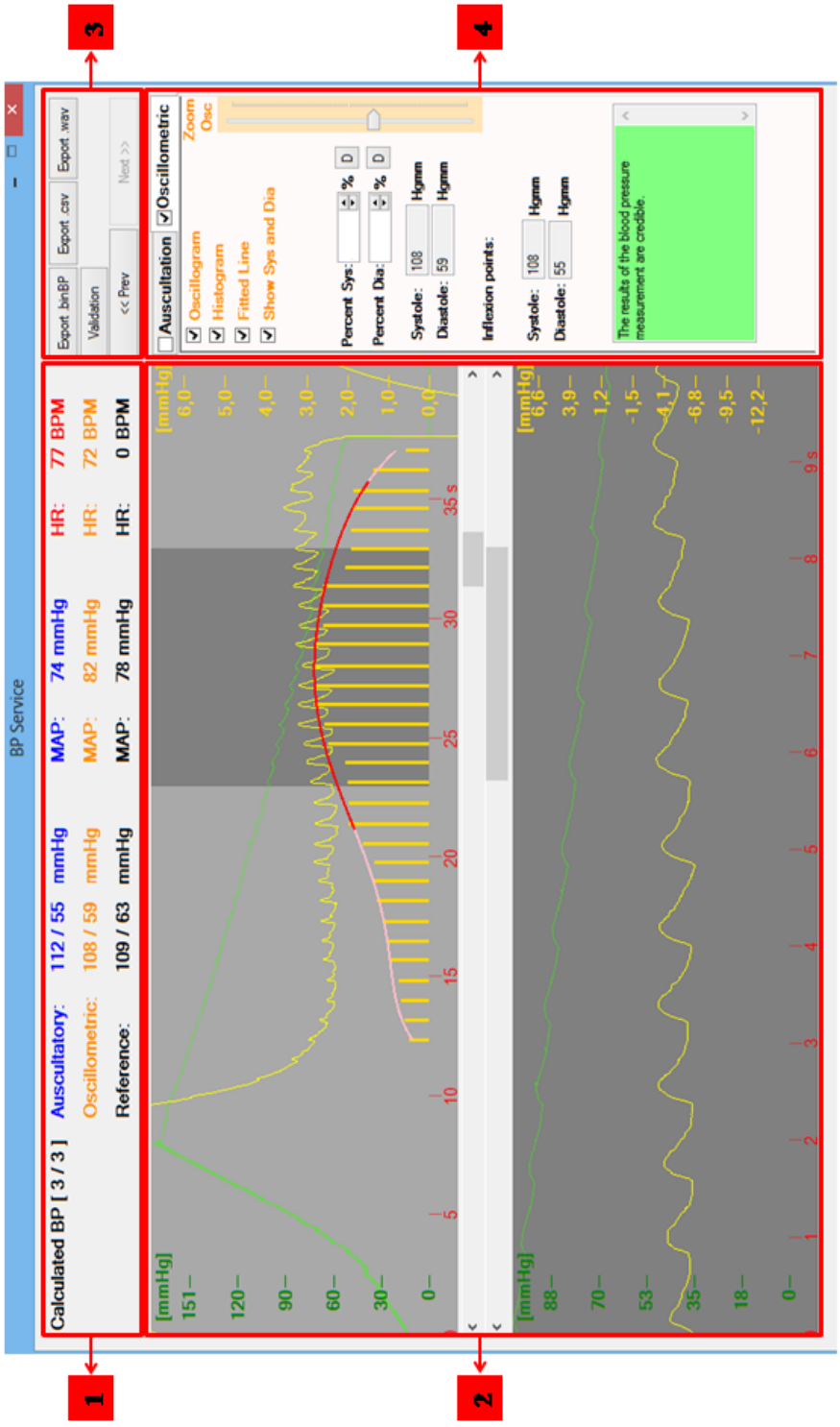

Figure 34: The parts of the visualization interface of Cardiospy BP Service

3.3.5.2.7 The panel for displaying the results

The panel displays the results of the blood pressure measurement. After the label of "Calculated BP" there is the serial number of the actual measurement of the recording and the number of the measurements in the recording. There are three rows which show the blood pressure values. The first blue row shows the results of the auscultatory results, the second yellow one shows the oscillometric results, and the third black one shows the manual reference values. All the three rows contain the SP, the DP, the MAP, and the heart rate. If the item of recording has no reference values the third black row contains 0 values.

3.3.5.2.8 The canvas

This unit, which has two panels, visualizes the data of each step of the blood pressure algorithm. The upper panel displays the data of the whole measurement from the beginning of the inflation of the cuff to the end of the deflation. The lower panel gives an enlarged picture of grey stripe of the upper panel. There are two scroll bars between the two panels. If the upper scroll bar is changed, the picture jumps to the next or the previous measurement of the recording. If the lower scroll bar is scrolled the grey stripe moves, so the user can navigate in the measurement to analyze the parts of the curves.

The canvas visualizes the following data:

- Cuff pressure: Its waveform is drawn by green color on both panels of the canvas.
- Oscillogram: it is drawn by yellow color on both panels of the canvas.
- Histogram: It is drawn by orange only on the upper panel of the canvas. The place of each item of the histogram is at the point of the local minimum of the oscillogram accurately. The height of each item of the histogram shows the size of the press change.
- Wrapping curve: It is drawn by pink on the upper panel of the canvas. The wrapping curve is a polynomial which fits to the points of the histogram.
- SP and DP: The systole and the diastole are visualized by red. The systole is represented by the beginning point of the red part on the wrapping curve, whereas the diastole is represented by the end of the red part. The algorithm calculates the SP and DP in both ways: height-based and slope-based. Our algorithm works better with the height-based method. Hence the application visualizes the height-based results.

On the two panels of the canvas there are scales. On the right side of the upper panel there is a yellow scale which belongs to the histogram. It has six grades with equal distances. On the right side of the lower panel there is an orange

scale which belongs to the oscillogram with eight grades. On the left side of both panels there is a green scale, which belongs to the cuff press with six grades. The scales help the user to read the values of the cuff press, the oscillogram and the histogram. The unit of the scales is mmHg. The scales are static, because the grades are fixed. The scales are dynamic, because the values of them are counted based on the values of each object.

The surface is interactive, which means that if the user changes something on the surface, the results are intermediately shown. If the user changes the zoom of the canvas with the Track Bar on the Control Page, the curve of the cuff press, the oscillogram, the histogram are dynamically redrawn and the values of the grads on the scales are also recounted. If the ratio of the height-based method changes, the SP and DP are recounted and the red part of the wrapping curve also changes.

In the point of the mouse there is a rectangle on the canvas. In the rectangle there are several pieces of information about that point, namely the cuff pressure, the time passed from the beginning of the inflation and the height of the oscillogram. If the mouse moves the values are recalculated.

On the canvas there is a red vertical line, which shows the exact place of the mouse horizontally. If the user moves the mouse the red line goes together with it. The line helps the user to find the points of the cuff press, the oscillogram, and the histogram are connected together. The user can examine the connection of the three objects.

3.3.5.2.9 The buttons

The user can navigate to the previous and the next measurement in the recording with the "Prev" and the "Next" labeled buttons.

3.3.5.2.10 The oscillometric control page

On this page the user can control the visualization of data of the oscillometric algorithm. The "Oscillometric" named Control Page is divided into five main parts as Figure 35 shows.

1. With four Check Boxes the user can turn the curves on the canvas off or on. Each Check Box represents the oscillogram, the histogram, the wrapping curve, and the red part of the wrapping curve representing the systole and the diastole.
2. The part of the Control Page shown by number 2 connects to the height-based method. There are two Spinners, which show the ratios of the height-based method. If the user changes these values, the SP and the DP are

refreshed. The range of the Spinners is 0 . . . 100. There are two Buttons next to the Spinners which restore the default ratios. In the two Text Boxes there are the SP and the DP calculated by the height-based method.

3. This part shows the blood pressure result of the slope-based method.
4. If the user changes the Track Bar the curves on the canvas are vertically enlarged or diminished.
5. The green panel gives information to the user about the results of the process of acceptability and the authenticity.

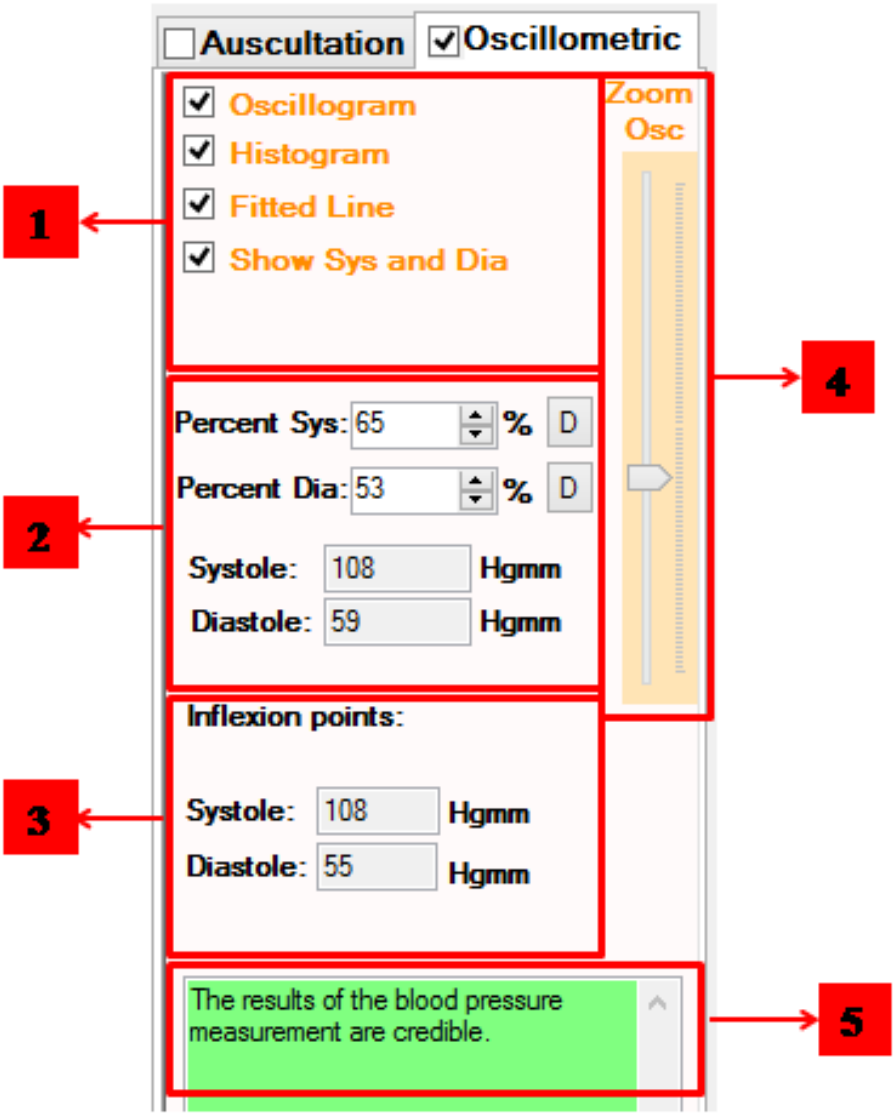

Figure 35: The oscillometric Control Page of Cardiospy BP Service

3.3.5.2.11 Acceptability and authenticity of the results

The algorithm gives information about the recording. If the recording is very noisy, the algorithm recognizes it. The algorithm considers a recording as a noisy one, if there are a lot of (more than a limit) bars in histogram whose height are greater than a parameter. In this case the algorithm may give results, but it can be useless and not authentic information. If the algorithm decides that the recording is very noisy the application shows a white, empty text on the Control Page, the SP and the DP are 0, and there is a big red sign in the right upper corner on the surface which warns the user that the recording is "INVALID".

The measurement of the recording can be incomplete, because the microcontroller has finished the recording earlier. In this case the algorithm tries to give results. If there is not enough data in the measurement, a big red flash sign appears in the right upper corner on the surface which shows that the measurement is a "BAD RECORDING".

If the algorithm gives results, it may be unbelievable. Lackovic (2003) says that the systole has to be in the range 50 and 280 mmHg, the diastole in the range 40 and 140 mmHg. The difference between them has to be at least 10 mmHg. If the result satisfies the previous condition on the Control Page the green panel gives information about it. If the result is near the limitation of the conditions or the wrapping curve is very different from the ideal on the Control Page the yellow panel informs the user that the result is questionable. If the result does not satisfy the conditions, the red panel appears and informs the user that the result is not acceptable.

3.3.5.3 Validation

In order that the BP Service application can be put to the market, its algorithm has to satisfy the international standards. The standards specify on what kind of and how many recordings the algorithm has to be executed. The results of the algorithm have to be compared to some reference values. The validation has to be performed on the mass of recordings. The results have to be analyzed on statistically. The standard specifies the acceptable statistical indexes.

If the user clicks on the "Validation" button in the BP Service the validation application is executed. The validation application is built to support the statistical analysis based on the standards of the British Hypertension Society (BHS) (O'Brien et al., 1993) (Kobalava et al., 2003) and the European Society of Hypertension (ESH) (O'Brien et al., 2010).

The application can work not only with one measurement, but a mass of measurements of the recordings, too. The application executes the blood pressure measurement application on each measurement. The results can be analyzed by the statistical tools of the validation application. The application is ready to analyze algorithms running on microcontrollers.

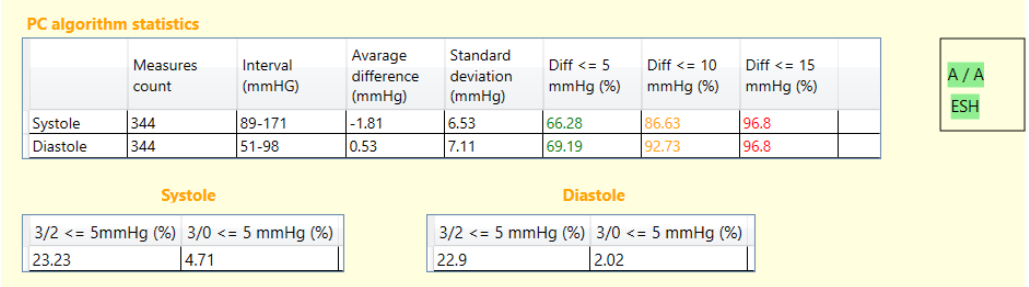

Figure 36: Validation tables

The application builds tables to examine whether the algorithm satisfies the requirements of the standards. Figure 36 shows the statistics window of the validation application.

There are the Bland-Altman plots on the other tab of the validation application see Figure 37.

These plots are built to analyze the difference between the results of an algorithm and the reference values (Bland and Altman, 1986) (Bland and Altman, 1999). Many articles use the Bland-Altman plots (Myers, 2010), (Aboy, 2011), (Lin, 2007).

The Bland-Altman plots show points in a coordinate system. The x value of the point represents the average of the algorithm value and the reference value. The y value of a point represents the difference between the algorithm value and the reference value.

The application creates Bland-Altman plots for both the systole and the diastole. On the surface there are buttons which help the user choose which values they want to analyze, the systole or the diastole.

In the validation application the user can change the parameters of the height-based algorithm. The validation application executes the blood pressure algorithm with default values (now this is 65% for the systole and 53% for the diastole) when the recordings are imported. If the user changes the default values, the

blood pressure algorithm is executed again on all the imported recordings and the values are refreshed on the validation surface.

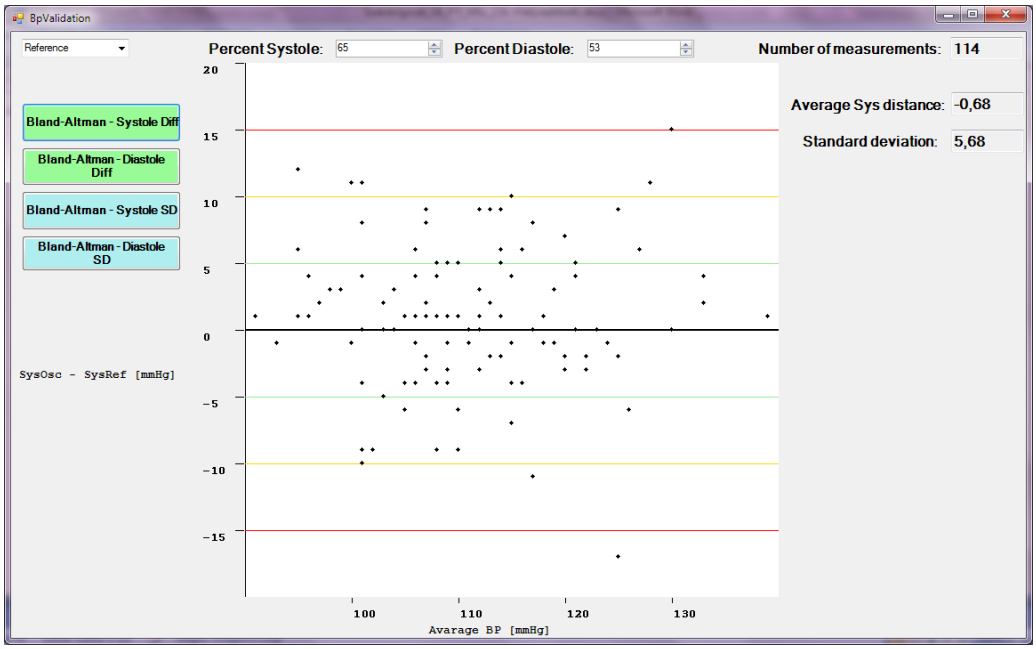

Figure 37: The Bland-Altman plots

3.3.6 Further work

We plans to improve our solution for blood pressure measurement. One plan is that we process and send blood pressure measurement recordings by mobile phone, which allows remote monitoring for the physician. Similar device is offered by the iHealth (iHealth, 2015).

Another plan is to use the algorithm during stress tests. Because the patient moves the signals will contain a lot of noise.

From the gathered signals a blood pressure measurement database can be built. The database can help to find the specialties of a patient, or to recognize what the differences are between the old recordings and the new recordings, which can enlighten an illness. Based on the database information about its population can be retrieved with data mining tools.

4 Education of database programming

In the education field you can also find intelligent data processing problems. If the teachers use an active learning method they have to verify every single solution of the students. But a student will give not only one solution to a task, which means that the teacher can have a lot of duties. A software application can support the duties of the teacher who uses an active learning method in organizing the students, the tasks, and the solutions, moreover, in following the performance of the students. In the case of education of programming the application can also help in the syntactic verification and it may also help in a kind of semantic verification.

I used the active learning method during the Advanced Database Management Systems 1 course at the Faculty of Informatics, University of Debrecen. It is one of the subjects related to the database systems for Software Engineering BSc students. The students learn advanced SQL and PL/SQL in Oracle environment. The course consists of a 100 minute lecture and 100 minute laboratory practice per week. In the lecture the students get acquainted with the features of the Oracle database management system (DBMS). In the laboratory they practice the new material which they have learnt in the lecture.

To support the active learning method in the subject, I planned and implemented a software application. In the implementation a student, József Kányási helped me.

4.1 Preliminaries

The lecture of Advanced Database Management Systems 1 is traditional. The teacher shows the material to the students, and they take notes. In the lecture room there are no computers, they cannot try out the programs. The main goal of the laboratory is that the students try out the features of the Oracle. In the laboratory many teachers use the traditional educational method, which means that the teacher works at the whiteboard or the projector and shows tasks with solutions to the students. With this method the students learn in a passive way, in most of the time they do not ask, they have no individual thoughts, they only sit and take notes. The teacher does not know whether they understand the solutions or not, whether they could solve the tasks alone or not. Sometimes the students do not give any feedback about their knowledge and problems. Maybe they try

out the program codes at home, and practice at home. The first feedback is the result of the test paper. But this is too late for the students, because they do not have enough time to correct the mistakes. The students most often do not ask questions about the problems, they learn something together that they think to be good.

4.2 Active learning method

The "learning by doing" or active learning method is well-known and applied in many fields in education.

The classic didactic authors, like Skinner (1986) and Polya (1957) say that the learning process of students is more efficient if they learn in an active way instead of a passive one. The students should solve tasks and problems, and acquire experience instead of observing the tasks and their solutions of the teacher. The tasks are to guide and help the students in the learning process. The "learning by doing" concept appear also in the new didactic books, like books of Schank et al. (1999) and Dufour et al. (2010).

4.3 Literature review

Although the new researchers often reject the old principle, the experimental learning is relevant nowadays. The concepts of Skinner (1986) and Polya (1957) can also be discovered in the recent didactic books. In their didactic book, Dufour et al. (2010) give an experimental education framework to the teacher. They show how the teachers can organize the work in the school and in the classroom. Roberts (2011) presents a very good overview and historical survey of experimental education in his book.

The "learning by doing" concept works also in the education of computer science. Gogoulou et al. (2009) used a software application for exploratory and collaborative learning in the education of programming. Drake (2012) deals with experimental learning, but he points out that the active learning is not proper for every educational situation. In the area of database systems Ramakrishna (2000) describes an experimental education survey of the undergraduate education. His results show that his students prefer the experimental learning over the traditional tutorials. Moore et al. (2002) describe a relational database management system course at Texas A & M University Corpus Christi that uses experimental learning. They receive a very good feedback from the participating students. Mason (2013) also presents experimental learning for teaching database administration and

software development at Regis University. His students indicated that the course was a successful experience that helps them fine-tune their technical skills and to develop new soft skills.

There are applications which help the teacher verify the solutions of the tasks like the evaluating tool of Kósa et al. (2005). Their application can examine C, Pascal, C++, and Java programs in such a way that the application executes the solutions for a lot of test cases, and if the result is the same as the predetermined output, the solution is correct. They do not check the program code itself, only the results. I cannot use this method, because an SQL or PL/SQL code results in many changes in the database.

4.4 Advanced DBMS 1 course

The Advanced DBMS 1 course follows the material of the book titled PL/SQL Programming written in Hungarian (Gábor and Juhász, 2007), and uses also the actual documentation of Oracle (Oracle Documentation, 2013). The syllabus of the course can be found in Table 14 and 15. Tasks of each laboratory practice can ask to use the programming tools of previous lessons.

1.	Main features of PL/SQL, PL/SQL block, constant, and variable declaration, basic types of PL/SQL, DBMS_OUTPUT.PUT_LINE procedure, statements
2.	Data types of PL/SQL, SELECT INTO statement, SQL DML with RETURNING
3.	Character set, lexical units, expressions in PL/SQL
4.	Subprograms in PL/SQL
5.	Exceptions and exception handling in PL/SQL, CURSOR FOR LOOP statements
6.	DML triggers
7.	System triggers, updatable and non-updatable views
8.	Cursors and cursor variables
9.	Transaction processing and control in PL/SQL, locking in PL/SQL
10.	Packages
11.	Composite types of PL/SQL 1.
12.	Composite types of PL/SQL 2.
13.	Native dynamic SQL
14.	Write efficient PL/SQL program
15.	Other topic of PL/SQL programming

Table 14: The syllabus of Advanced DBMS 1 lecture

1.	Practice of SQL
2.	Tasks using blocks, declarations, statements, and the DBMS_OUTPUT.PUT_LINE procedure
3.	Tasks about data types of PL/SQL, SELECT INTO statement, SQL DML with RETURNING
4.	Tasks of expressions in PL/SQL
5.	Tasks of subprograms in PL/SQL
6.	Tasks of exceptions and exception handling in PL/SQL, CURSOR FOR LOOP statement
7.	Tasks of DML triggers
8.	Tasks of updatable and non-updatable views
9.	Tasks of cursors
10.	Tasks of cursor variables
11.	Tasks of packages
12.	Tasks of composite types of PL/SQL 1.
13.	Tasks of composite types of PL/SQL 2.
14.	Tasks of native dynamic SQL
15.	Writing examination paper done under supervision

Table 15: The syllabus of Advanced DBMS 1 laboratory practice

The prerequisite courses are the Database Systems and the Programming Language 1 (Bulletin of Software Engineering BSc, 2007). This means that the students who attend Advanced DBMS 1 course, have learnt SQL and programming in the language C. So the teacher of Advanced DBMS 1 can rely on the fact that the students can write SQL statements and programs in C.

The students get a signature at the end of the semester if they fulfill the requirements of the laboratory. If the students have a signature, they can take an exam in the exam term. If they do not pass the exam, they can resit it 2 times. On the exam the teacher asks theoretical questions, like definitions of concepts, how the features (for example trigger, loop, and cursor) work, what the proper syntax of a feature is, and so on.

4.5 Laboratory environment

The university has a standalone server on which an Oracle Database 11g is installed. Every student has a personal database account. A schema belongs to each database account. In this schema the students can do everything they want. (Of course the database administrator gives privileges to students, but not all the privileges. So they can do everything for which they have privilege.) In this schema, the students can explore how they can use the new material. The

students can connect to the database from every computer which has internet connection.

The students use SQLDeveloper as a client program. It is a very simple program, it needs no installation. The students copy it to the computer and they can use it.

4.6 The active learning method

My idea is that I teach the Advanced DBMS 1 course in a way that I as a teacher give tasks to the students, so they practice SQL and PL/SQL alone, they show the solutions to the teacher, and the teacher gives feedback to them. In this way the students learn in an active way. (Vágner, 2014)

I have organized the work in the course so that it meets the following requirements:

1. In the laboratory the students work independently. I give tasks to the students every lesson, and they solve it.
2. I give some feedback for every solution of each student.
3. If the solution is wrong, the students can give a new solution for the task before the deadline.
4. The students work on the tasks in the lessons, but if they do not finish the tasks, they can continue at home. If they complete the tasks in the lesson, they have no homework.
5. The tasks have deadlines. Because the students can work on the tasks at home, the deadline cannot be earlier than the end of the next lesson. If the students solve a task at home, they get the feedback on the next lesson. If the solution is wrong, they can give proper solution on the next lesson.
6. The students work independently, but they can discuss the tasks with each other.
7. The laboratory is based on the lecture; the students practice the new material of the lecture. In the laboratory there is no new material.
8. I as a teacher have to organize the lecture in a way that the students can work alone with the new material in the laboratory. The syllabuses in Table 14 and Table 15 show that in the lecture there are a few topics which the students cannot practice or which can be practiced without other topics. These topics are: types, character sets, lexical unit, locking, system triggers, etc. So, in the lecture the teacher should introduce materials which the students can practice in the laboratory.

Drake (2012) finds problems with active learning, and he says that it can be used under given circumstances. In the case of Advanced DBMS 1 the student has

previous knowledge about programming and database systems. They take part in the lectures, which show the new material and examples. So, first they get to know the material then they practice it. In the guidance the teacher has to find the golden mean: not too much but not too little. The first guidance is the lecture; the second are the tasks, which are getting more and more difficult during the lessons and the semester. The third guidance takes place in the laboratory where some students need more explanation, some students need nothing. My goal is to give feedback for all activities of the students; this means the laboratory activities and the homework (if there is). The interested students often write me emails with questions, and we have time to make conversations in the laboratory. The first goal of my active learning method is to enable the students to use the PL/SQL and SQL as a skill, namely they will get a practical competence which can be immediately used in business.

4.7 The software application for supporting the education of database programming

I and one of my students, József Kányási created a software application framework to support the active learning method. The idea, the design, and the model was mine and I implemented the most part of its database side. It has two main parts: the database objects and a C# program. (Vágner, 2015)

The first goal of the software system is to administrate the tasks given to the students and the solutions made by them. The second one is to check whether the syntax of an uploaded solution is correct or not.

The administration helps both the teacher and the students to follow to performance of the students. The syntactic verification helps the teacher to give feedback. It is not easy to find syntax errors in the program code without execution. The system throws an exception when a solution with incorrect syntax is uploaded. The teacher can set deadlines for the tasks with the application, which can also enforce these deadlines.

4.7.1 Database objects

The database objects implements a submission system to help the work in the laboratory. A similar system is used by Kósa et al. (2004) for contests. Their system evaluates the solutions, but it does not check how the competitor solved the tasks. My goal was to administrate the tasks and solutions, check the syntax, but the semantics evaluation is done by the teacher.

The tasks and the solutions are also stored in the database. There is a schema named ADBMS, where tasks and solutions are stored. Figure 38 shows the tables of the ADBMS schema and the relationships between them.

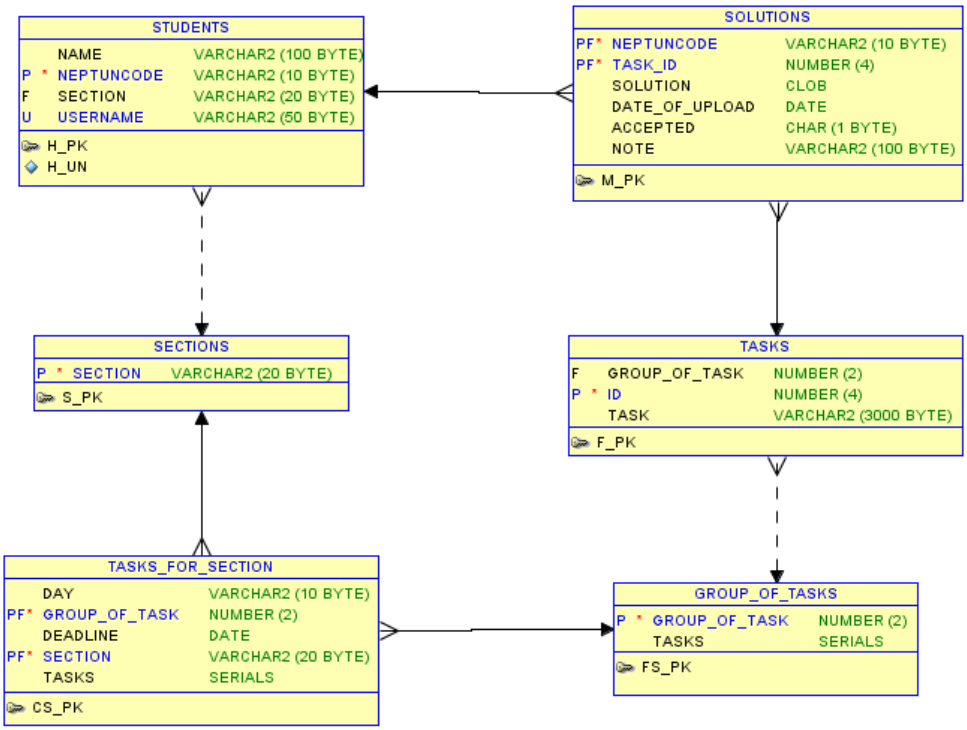

Figure 38: ADBMS schema

The STUDENTS table stores the students who attend the course. The students attend this course in more than one section in a semester. Basically, the schedules of the sections are the same in each semester, but it can occur that there is a difference between the numbers of the lessons because of public holidays. The SECTION attribute helps create groups of students for each section in a semester. The TASKS table stores the text of the tasks. It is worth dividing the tasks into groups because there are at least 3 tasks in one lesson but this number can be 10 or more. The GROUP_OF_TASKS table organizes the tasks into groups. The TASKS_FOR_SECTION table assigns the tasks to the students. The students in each section get the same deadline for each group of tasks. The SOLUTIONS table stores the solutions of the students. It also stores the date when the student uploaded the solution, a flag which shows whether the solution is accepted by the teacher or not, and a note.

Each student can only read the data from the ADBMS schema, but not everything. This means that a student gets privilege only for selecting two views in the schema: MYTASKS and MYSOLUTIONS. The result of selecting the MYTASKS view contains the tasks (the identifier and the text of the task and its deadline) which the student has to solve. This is a subset of the rows in the TASKS table. The students can check their own personal uploaded solutions in the MYSOLUTIONS view. This view contains not only the accepted solutions but all uploaded solutions. The student can check in this view whether the solution of each task is accepted, has to be corrected, or is waiting for examination. In the view there is the comment column. The teacher writes comments into this column if the student does not participate in the lesson and the solution of the task is not accepted. The comment shows the teacher and the student what the problem is with the solution and what has to be corrected.

A student can upload a solution into the database by executing the UPLOAD_SOLUTION procedure. The students have execution privilege for the UPLOAD_SOLUTION procedure. The procedure has two parameters. The first parameter is the identifier of the task; the second parameter is the text of the solution. The procedure automatically verifies the syntax of the solution. The UPLOAD_SOLUTION procedure also checks the deadlines and throws an exception for the solutions where the deadline of the task has expired.

4.7.2 Syntactic verification of the solutions

The UPLOAD_SOLUTION procedure performs the syntactic verification. The students often make a mistake in the program code and may be too lazy to execute their solutions. Without syntactic verification it can happen that a solution is accepted by the teacher, but it contains syntax errors.

The database management system can perform the syntactic verification automatically. The solution which the student wants to upload is executed by the UPLOAD_SOLUTION procedure in the schema of the student. The procedure throws an exception if the program code cannot be executed, and in that case the solution will not be uploaded. The students are obliged to correct the syntax errors in their work.

The Oracle PL/SQL has a very good tool named native dynamic SQL, which I used to realize syntactic verification. The native dynamic SQL can execute an SQL statement without semicolon at the end or a PL/SQL unit.

In a solution the students can upload more than one statement or PL/SQL unit. It does not matter whether each of them is SQL or PL/SQL code, however, at

the end of the SQL code the students cannot use semicolon. Every statement or PL/SQL code has to be closed with a "/" sign. The UPLOAD_SOLUTION procedure splits the uploaded solutions into SQL statements and PL/SQL units based on the "/" sign. Then it executes the parts of the solutions one after the other in the schema of the student who executed the UPLOAD_SOLUTION procedure. If one of them throws an exception, the others will not be executed. This also means that the whole solution is wrong, so it will not be uploaded to the ADBMS schema.

The students have to make sure that the statements work in their schema. Let us see an example. If a student wants to upload a CREATE TABLE solution and the name exists in their schema, the uploading of the CREATE TABLE statement will give error despite the fact that the statement is correct. So the student has to drop the table first or has to choose another table name.

Moreover, the student may clean their schema because the UPLOAD_SOLUTION procedure, which executed their codes in their schema, can make a lot of schema objects, like tables, views, procedures, etc.

4.7.3 The application for supporting the teacher

The application which supports the tasks of the teacher has the following features:

1. Imports or inserts students and sections into the ADBMS schema. The data of the students can be changed with the application.
2. Publishes the text of the tasks. This means that the teacher writes the text of the new tasks and arranges the tasks into groups. Of course, later the text of the tasks can be modified. A task can be deleted until a student gives a solution to this task. The deadline of each group of tasks can also be determined. Of course the deadline can also be modified.
3. Helps the teacher verify the solutions which are uploaded by the students. This feature is the main reason why the application was developed. The program can list the text of the solutions with the name and the section of the student, the identifier of the task, and identifier of the group of tasks. The program also shows for each solution whether the solution is accepted, refused, or the teacher has not yet dealt with it. The program can filter and order the rows by these values. This list is very long at the end of the semester, so filtering is very important. When a teacher chooses a solution from the list, the application shows the text of the solution, the text of the task, and the name of the student in another window. This window can be easily read. In this window the teacher can set whether the solution is accepted or refused. The teacher can also add a comment to the solution in

this window. The teacher can do the semantic verification of the solution with the help of this window.

4. Lists the students who have not uploaded solutions for the given tasks. With this feature the teacher can easily find the students who are not ready with each task.
5. Lists the names of the students and shows how many per cent of all the tasks they have solved. The teacher can find the students who do not work hard enough in the semester or who have completed all the tasks.
6. Lists the name of students, the groups of tasks, and shows how many per cent of the tasks in the group are not solved by the student. With this function the teacher can check whether a student worked in the lessons or not.

4.8 Evaluation of the application the active learning method

I used the active learning method for 3 years. In the last year I had control groups with traditional education. (Vágner, 2014)

4.8.1 Evaluation of the 2009/2010 spring semester

This was the first year when I used the active learning method. The requirements of the laboratory had two parts. One of them was that the students had to solve 75 per cent of the given tasks. The other part of the requirements was that the students had to write two test papers and they had to achieve 60 per cent of the maximum score of the two test papers. The students could use only a pen when they wrote the test paper; there was no computer and no documentation. If both requirements were fulfilled, the student got the signature.

In this year I did not give any deadline. A lot of students said that they would do the tasks the next week. At the end of the semester they wanted to finish the tasks. This was a hard job for them and also for me. The students gave solutions right before the last test papers, so I could not check in time who had enough solutions to write the test paper. So I promised that next year there would be deadlines and that I would be strict in enforcing the deadlines.

The application was not ready. So I used SQL statements to verify the solutions. I spent a lot of time writing those SQL statements.

There was no syntactic verification in the UPLOAD_SOLUTION procedure. Some students uploaded solutions which did not work. And I did not find every syntactic problem in the solutions. These students did not want to try out the solutions.

There were 5 sections and 92 students. 48 students got signature. There were 2 students who solved all the tasks and gave proper solutions, but could not achieve enough points on the test paper. The reason was that they copied the code from the documentation, and they did not remember the exact syntax without the documentation. The other students did not give enough proper solutions in the laboratory.

4.8.2 Evaluation of the 2010/2011 spring semester

The requirements of the laboratory were the same as in the previous year. In this year I gave deadlines, and I was strict in enforcing them. The application was ready. I checked the solutions in the application and I could check whether each student works or not. If the student did not work in the lesson, I could ask why. I could warn the students if they were behind with the solutions.

There was a syntactic verification in the UPLOAD_SOLUTION procedure. The students had more difficulties when they uploaded the solutions, because it threw an exception if the solution could not be executed. My job was easier, because I had to take care only of the semantics of the solutions.

There were 5 sections and 73 students. There were 15 students who solved all the tasks and gave proper solutions. They tried to be perfect. There were again 3 students who solved all the tasks, gave proper solutions, but could not achieve enough points on the test paper. 51 students got signature. The other students did not give enough proper solutions in the laboratory or did not write the second test paper.

4.8.3 Evaluation of the 2011/2012 spring semester

In this year I wanted to compare the results of the traditional teaching method to the active learning method. There were 5 sections. In 2 sections I used the traditional teaching method; in the other 3 sections I used the active learning method.

The traditional teaching method means here that on the whiteboard I showed tasks and theirs solutions, which were related to the previous lecture. After the second week the students had to write a ”small” test paper with one or two tasks

every week. The first requirement of the laboratory was that the students had to give proper solutions for 60 per cent of the given tasks of the "small" test papers. There were 11 "small" test papers in the semester.

The active learning method was the same as in the previous years. The first requirement of the laboratory was that the students had to solve 80 per cent of the given tasks.

If the students accomplished the first requirement, they could write the test papers at the end of the semester. They had to achieve 60 per cent of the maximum score of the test paper. The students could use only a pen when they wrote the test paper; there was no computer and no documentation.

In the sections where I used the traditional teaching method, there were 27 students. 10 students gave up the semester after a few "small" test papers. This means that they wrote 5, 6, or 7 "small" test papers, and they did not attend the course any more. 14 students wrote the final test paper. 13 students got signature at the end of the semester. So I can say that the students who were learning continuously in the semester wrote successful "small" test papers, and the final test paper was easy for them.

In the 3 sections where I used the active learning method, there were 36 students. 4 students could not solve 80 per cent of the given tasks. They were not allowed to write the final test paper. 21 students got signature at the end of the semester.

4.8.4 Comparing the two teaching methods based on the performance of the students

On the first lesson 60 students wrote a test paper about their practical knowledge of programming and database systems. They had to write C program codes and SQL statements. Table 16 shows the result of this test paper.

The results of the test paper were similar in both cases. We can see from the table that the AL students (the students who took part in the active learning method) got more signatures than the TT students (the students who took part in the traditional teaching method).

	TT students	AL students
Number of students	26	34
Got signature	50%	61%
Average result of programming part	50,64%	51,96%
Average result of SQL part	25%	23,39%

Table 16: Results of the test paper of the first lesson

I performed a hypothesis test, a one-tailed two-proportion z-test (Freund and Wilson, 2003) to determine whether the difference between two proportions is significant or not. My goal was to verify that the active learning method is more effective, namely more AL students got signature than TT students. The null hypothesis is $H_0 : p_1 = p_2$, where p_1 is the proportion of AL students who got signatures, whereas p_2 is the proportion of TT students who got signatures. I tested this null hypothesis against the one-sided alternative hypothesis $H_1 : p_1 > p_2$ that is the probability of an AL student getting a signature is greater than a TT student getting one. Let n_1 denote the number of AL students and n_2 denote the number of TT students. In this case $p_1 = 0,61$; $n_1 = 34$; $p_2 = 0,5$; $n_2 = 26$.

The pooled sample proportion is

$$p = \frac{p_1 n_1 + p_2 n_2}{n_1 + n_2} = 0,56223$$

The standard error is

$$SE = \sqrt{(p(1-p)\Big(\frac{1}{n_1} + \frac{1}{n_2}\Big)} = 0,1292$$

The test statistic is

$$z = \frac{p_1 - p_2}{SE} = 0,8511$$

The P-value is the probability that the z-score is greater than 0,8511. Since the test statistic is a z-score, I used the normal distribution to assess the probability associated with the z-score. So the P-value is 0,1981. If the significance level is 0,2 (80%), the result is statistically significant.

If you want to accept the result when the desired significance level is 0.05 (95%) with the same proportion, I have to examine the difference between the two teaching method on about 230 students with nearly equal group sizes. If you want to accept the result where the desired significance level is 0.1 with the same

proportions I have to examine the difference between the two teaching methods on about 140 students with nearly equal group sizes.

Table 17 shows the average of the exam marks of the students. 31 students took the exam in this semester. There is no significant difference between the results of the two groups. Based on the table we can say that the exam mark did not depend on the teaching method of the laboratory.

	TT students	AL students
Number of students	11	20
Average of exam marks	2,54	2,45

Table 17: The average of the exam marks of the students

On the exam the teacher asks theoretical questions instead of practical tasks. This way the better practical knowledge does not influence the exam mark. The students who are not so good in laboratory can swot up the concepts and other theoretical questions and get a good mark. On the other hand, the students who are very good in practice may not learn everything for the exam.

4.8.5 Voluntary survey of the students

I asked the AL students about the active learning method. I got 63 answers. 71% of the respondents passed the exam. 90% of the respondents liked that they had to solve the tasks individually in the laboratory. 80% of the respondents would not have liked that the teacher solved the tasks in the laboratory. 14% of the respondent did not mind it. 54% of the respondents would like the active learning method to be applied on other courses. For 19% of the respondents it makes no difference. 75% would choose the active learning method in the laboratory of the Advanced DBMS 1 instead of the traditional style. 12% of the respondents do not mind it. 7% of the respondents said that they uploaded solutions without trying it out.

The students gave other individual feedback. Here are the most important opinions:

- It was a good experience to solve the tasks alone.
- I could not solve the tasks week by week.
- I liked this teaching method because I had to think about the solutions of the tasks. I did not need to follow the thinking of the teacher.
- Every student could work in an individual pace.

- Sometimes there were a lot of tasks for a lesson. I had to solve the tasks at home, and I got feedback only on the next lessons. So I had to work with the tasks of previous lessons instead of the new tasks.
- I understood the tasks better thanks to the individual work.
- I was forced to understand the usage of the material in the laboratory. I think that the active learning method should be used at most courses. The system was strict, but I did not feel that it goes against the students.
- Consistent, precise, and strict requirements.
- I would have liked if the teacher had given more solutions and comments, and we had been thinking together. I am bored of working alone.
- I did not attend the lectures, so I would have liked to get a short summary at the beginning of the laboratory.
- The active learning method was very good in improving of programming skill and solving problems.
- I liked that I could work alone, and if I had a problem, I received assistance.
- The students were forced to solve the tasks. It is not enough to monitor how they can be solved.
- I liked the active learning method. I got answers to all of my questions.
- The students had to discover how the features of the DBMS work. I understood the solutions better, and I remember the material better. I got help if I had problems.

I asked the TT students about the active learning method. I got 10 answers. 50% of the respondents would have liked to attend the section applying the active learning method. A student liked when I gave more than one solution for a task.

This result is similar to the ones of Ramakrishna (2000) and Mason (2013).

5 Summary

This thesis consists of 3 main parts. All of them are related to intelligent data processing.

In the first part I introduce data mining algorithms namely clustering techniques. In this part I present a new clustering technique which can be considered as a combination of two well-known methods.

The second part is about the biomedical signal processing, namely ECG and blood pressure measurement signal processing. In this part two new modules of Cardiospy of Labtech Ltd. are demonstrated.

In the third part I present how I used the learning by doing teaching method in a subject whose topic is database programming. I summarized results of the teaching method which I used for 3 years. When applying the teaching method I used a software application which supported the managing of the tasks and the solutions of the students. Its syntactic verification helped the teacher in the verification of the solutions.

5.1 Clustering algorithm

In Chapter 2 several well-known clustering algorithms are described, such as OptiGrid, CLIQUE and Wavecluster, which are grid-based techniques, and DENCLUE, DBSCAN and OPTICS, which are density-based techniques. The concepts of the DBSCAN techniques are explained, such as ϵ-neighborhood of an object, core object, $MinPts$, directly density-reachable, density-reachable, and density-connected. The OPTICS is different from the DBSCAN in a sense that it builds a reachability plot giving more than only one cluster set as a result. From the reachability plot many results (each of them is a cluster set) can be read. It is also explained how the clusters can be read from the reachability plot which is the result of the OPTICS algorithm. Then a lot of publications of the combinations of the grid-based and the density-based techniques are summarized.

Afterwards, a new clustering algorithm named GridOPTICS (Vágner, in press) is introduced which is a combination of a grid clustering and the OPTICS algorithm. It builds a grid structure from the input points in the first step, then it executes the OPTICS algorithm on the grid structure in the second step, afterwards, it determines the clusters of the grid points in the third step, and finally, it assigns the input points to the clusters.

The GridOPTICS algorithm reads the clusters from the reachability plot in a different way as the OPTICS. This technique is also described.

The GridOPTICS algorithm has the same advantage as the OPTICS, namely it builds a reachability plot to find more than one clustering structure. Moreover, its execution time can be faster with more orders of magnitude than the OPTICS, which is very useful for large data sets which have more thousand points. However, the GridOPTICS can be less accurate than the OPTICS. The user of this algorithm has to decide whether the accuracy or the speed of their application is important.

In the experimental results a comparison is made between the execution times and results of the GridOPTICS and the OPTICS. Both algorithms are executed on several data sets which have various cardinality between 400 and 8000. More examples are given for data sets clustered by the GridOPTICS. Finally, suggestions are also given when to use the GridOPTICS algorithm and how its parameter can be chosen.

5.2 Biomedical signal processing

In Chapter 3 two new modules of Cardiospy of Labtech Ltd. are introduced. I worked together on these two applications with István Juhász, László Farkas, Péter Tóth, and 4 students of the university, József Kuk, Ádám Balázs, Béla Vámosi, and Dávid Angyal.

5.2.1 ECG signals

In Chapter 3.2 it is described what ECG is, how signals can be produced, and what the meanings of this signals are. Then a description is given how the analysis of the ECG signals can be automated by algorithms.

Then the new module of Cardiospy is introduced which clusters and visualizes the long (up to 24-hours) recordings of ECG signals. The model replaces the manual evaluation of long recordings, which is a lengthy and tedious task. The module put seemingly similar heartbeats into one group. Thus, cardiologists do not have to examine all (often more than 100000) the heartbeat curves to find the heartbeats which morphologically differ from the normal beats. They only have to analyze groups belonging to abnormal beats. In the module, there are automatic and manual clustering features. We developed the clustering and visualization module in C# 4.0 programming language, in Visual Studio 2010 environment. (Vágner et al., 2011 A), (Vágner et al., 2011 B), (Juhász et al., 2009)

The input data of the algorithm is a three channel ECG recording. We also get the annotations and types of the heartbeats as input data. The algorithm splits the ECG signal into signals each of which belongs to a heartbeat based on the annotation. Then we characterize a heartbeat with a pair of two-dimensional points. We retrieve the point pair with the help of wavelet transformation. On the point pairs an early, special version of the GridOPTICS clustering algorithm is executed, because the OPTICS was too slow to handle this problem, moreover, we recognized that many points in the point set have the same coordinates or they are close to each other. The main reason that many points have the same coordinate is that the dispersion of points representing different types of heartbeats is not entirely random. As a result of the algorithm, clusters of heartbeats appear.

Afterwards, the interactive visualization interface is introduced.

The module allows the user to divide certain clusters into further groups manually.

The feedback of the users proves our measurements that show that this method efficiently supports the evaluation of HOLTER ECG.

5.2.2 Blood pressure measurement

In Chapter 3.3 a short summary is given of the types of blood pressure measurement techniques. The main focus is on the oscillometric method. There is a description of what kind of errors can happen during the oscillometric blood pressure measurement. Then the importance of the long-term ambulatory blood pressure measurement is introduced.

Afterwards, the new module of the Cardiospy system of Labtech Ltd. is introduced which realizes the PC-side processing of the oscillometric blood pressure measurement recorded by a microcontroller. The microcontroller has only limited memory and processor capacity. Therefore the microcontroller can produce inaccurate results or even no results. But the PC-side application can process the measurements in a more accurate way. The cardiologist or the researcher can analyze the steps of the processing of blood pressure measurement using the application. It helps recognize the errors emerged during the measurement. (Vágner et al., 2014)

The main steps of our algorithm are:

1. The algorithm splits the recording into measurements. It processes only one measurement at a time.
2. It creates an oscillogram based on the measurement using a band-pass filter.

3. It finds the local minimum and maximum points and the values of the oscillogram.
4. It creates a histogram from the local extrema. The histogram shows the change of the cuff pressure at a minimum point.
5. The algorithm fits a wrapping curve which is a polynomial to the histogram.
6. It determines the maximum point and its place of the wrapping curve.
7. It determines the blood pressure values.
8. It gives information whether the result is acceptable or not.

The application visualizes the results of each step of the processing in an interactive way. Namely, if the user changes the parameters, the scale or the position of the mouse, the objects on the surface change immediately. The interactive visualization surface helps the user understand the information of blood pressure measurement better. The application is mainly built for processing long recordings including more measurements. Most recordings are recorded during 24 hours. The application easily navigates among the measurements of one recording.

We have built a validation application to support the validation of the blood pressure measurement algorithm. The validation application can process a mass of the measurements with reference and visualizes the statistical data and the Bland-Altman diagram about the difference between the results of the algorithm and the reference data.

5.3 Education of database programming

In Chapter 4 the ”learning by doing” or the active learning method is introduced which is applied in a subject whose topic is database programming. The ”learning by doing” or active learning method is widely used and working properly also nowadays. In the education of programming it is very important that the students practice independently. The active learning method forces them to write program code and use the database management system independently. The task of the teacher is to guide and help the students through the material. The teacher gives personalized answers to the students, motivates them, and discusses the problems and the solutions. (Vágner, 2014)

To support the active learning method in the subject, a software application can be used which helps in organizing the students, the tasks, and the solutions, moreover, in following the performance of the students. In the case of education of programming the application can also help in syntactic verification and it may also help in a kind of semantic verification. (Vágner, 2015)

First the related literature is reviewed. Then the course is introduced. The topic of the course is Oracle SQL and PL/SQL. The laboratory environment is also described.

Afterwards, it is explained how the work of the lessons is organized and what the requirements of the used active learning method are, namely:

1. The students have to work independently on the given tasks.
2. Some feedback is given by the teacher for every solution of each student.
3. If the solution is wrong, the students can give a new solution to the task before the deadline.
4. The students work on the tasks in the lessons, but if they do not finish the tasks, they can continue them at home.
5. The tasks have deadlines.
6. The students work independently, but they can discuss the tasks with each other.
7. The laboratory is based on the lecture; the students practice the new material of the lecture. In the laboratory there is no new material.
8. The teacher has to organize the lecture in a way that the students can work alone with the new material in the laboratory.

Then the software application is presented which supports the education of database programming. It has two main parts: the database objects where the tasks and the solutions are stored and a C# program which can be used by the teacher to give feedback to the students. One of the main parts of the application is the syntactic verifier which refuses a solution if it cannot be executed, so the students can upload only syntactically correct solutions.

At the end of chapter 4 the evaluation of applying the active learning method is described. The experiments are gathered for three years. The results of the students show us that the laboratory results are better if the teacher uses the active learning method. The results of voluntary survey show us that students liked the active learning method; moreover, they would welcome it in case of other subjects also. The students have to work hard during the semester, but at the end of the semester they have experience in using the database management system, and they will not forget it for many years.

6 Összefoglaló

A disszertáció 3 fő részből áll, melyek mindegyike kapcsolódik az adatok intelligens feldolgozásához.

Az első rész adatbányászati algoritmusokkal, pontosabban klaszterező módszerekkel foglalkozik. Ebben a részben egy új klaszterező algoritmust mutatok be, amely két jól ismert módszer kombinációjának tekinthető.

A második rész az orvosi jelfeldolgozásról szól, pontosabban EKG jelek és vérnyomásmérés jeleinek a feldolgozásáról. Ebben a részben a Labtech Kft. Cardiospy rendszerének két új modulját mutatom be.

A harmadik részben azt mutatom be, hogy a "learning by doing" oktatási módszert hogyan alkalmaztam egy adatbázis-programozás témájú tantárgy során. A rész 3 év oktatási eredményeit összegzi. Az oktatási módszer támogatására egy szoftveralkalmazást használtam, amely a feladatoknak és a hallgatók megoldásainak a kezelésében segített. Az alkalmazás szintaktikai ellenőrzője segítette a tanári munkát, azaz a feladatok kiértékelését.

6.1 Klaszterező algoritmus

A 2. fejezetben néhány jól ismert klaszterező algoritmus kerül bemutatásra, mint az OptiGrid, a CLIQUE, és a Wavecluster, amelyek rács alapú technikák, illetve a DENCLUE, a DBSCAN, és az OPTICS, amelyek sűrűség alapú technikák. A fejezetben a DBSCAN algoritmus fogalmait mutatom be, azaz egy objektum ϵ-sugarú környezetét, a magobjektumot, a $MinPts$-t, a sűrűség alapon közvetlenül elérhetőséget, a sűrűség alapon elérhetőséget és a sűrűség alapon összekötöttséget. Az OPTICS algoritmus különbözik a DBSCAN-tól, mivel egy elérhetőségi térképet épít ahelyett, hogy csak egy klaszterhalmazt adna eredményül. Az elérhetőségi térképről sok eredményül kapott klaszterhalmazt lehet leolvasni. A fejezetben bemutatásra kerül, hogyan lehet leolvasni a klasztereket az OPTICS algoritmus eredményeként kapott elérhetőségi térképről. A fejezetben néhány olyan publikációt foglalok össze, amelyek a rács alapú és a sűrűség alapú technikák kombinációval foglalkoznak.

Ezután egy új klaszterező algoritmus, a GridOPTICS (Vágner, in press) kerül bemutatásra, amely egy rács alapú klaszterező és az OPTICS algoritmus kombinációja. Az algoritmus először az input adatokból felépít egy rácsstruktúrát, majd a második lépésben ezen futtatja az OPTICS algoritmust. Ezután a

harmadik lépésben meghatározza a rácspontok klasztereit, végül az eredeti pontokat hozzárendeli a klaszterekhez.

A GridOPTICS algoritmus nem az OPTICS-ban bemutatott módon olvassa le a klasztereket az elérhetőségi térképről. A dolgozatban bemutatásra kerül, hogy az algoritmus ezt pontosan hogyan végzi.

A GridOPTICS algoritmusnak megvan az OPTICS minden az előnye, azaz ugyanúgy felépíti az elérhetőségi térképet, hogy eredményképp ne csak egy klaszterstruktúrát kapjunk. Ezen kívül a GridOPTICS több nagyságrenddel gyorsabban futhat, mint az OPTICS, ami több ezer pontot tartalmazó nagy adathalmazoknál hasznos lehet. Azonban a GridOPTICS nem feltétlenül olyan pontos, mint az OPTICS. Az algoritmus használójának kell eldönteni, hogy az alkalmazásánál a pontosság vagy a sebesség a fontosabb.

A kísérletekben összehasonlításra kerültek a GridOPTICS és az OPTICS eredményei és a futási idejeik. Mindkét algoritmus lefutott olyan adathalmazokon, melyeknek a számossága 400 és 8000 közé esett. A GridOPTICS algoritmus eredményeire más példák is bemutatásra kerültek. Végül néhány tanács szerepel a fejezetben arra vonatkozóan, hogy a GridOPTICS algoritmust mikor érdemes használni és a paramétereket hogyan érdemes választani.

6.2 Orvosi jelfeldolgozás

A 3. fejezet a Labtech Kft. Cardiospy szoftverének két új modulját mutatja be. A két alkalmazáson Juhász Istvánnal, Farkas Lászlóval, Tóth Péterrel, és 4 egyetemi hallgatóval, Kuk Józseffel, Balázs Ádámmal, Vámosi Bélával, és Angyal Dáviddal dolgoztam együtt.

6.2.1 EKG jelek

A 3.2. fejezet leírást ad arról, hogy mi az az EKG, hogyan készül az EKG jel, mit jelent az EKG jel. Ezután bemutatom, hogy az EKG jelek elemzését hogyan lehet algoritmusokkal automatizálni.

Majd a Cardiospy új modulját mutatom be. A modul célja, hogy klaszterezze és megjelenítse a hosszú (akár 24 órás) EKG jeleket tartalmazó felvételeket. A modul felváltja a hosszú felvételek manuális kiértékelését, amely tevékenység egy hosszú és unalmas feladat. A modul a hasonló formájú szívütéseket egy csoportba sorolja, így a kardiológusnak nem kell minden (gyakran több mint 100000) szívütést ábrázoló görbét végigvizsgálnia, hogy megtalálja azokat a szívütéseket, amelyek

morfológiailag különböznek a normál szívütésektől, hanem csak a rendellenes szívütéseket tartalmazó csoportokat kell végigelemezni. A modul az automatikus klaszterezés mellett manuális klaszterező eszközt is tartalmaz. A klaszterező és vizualizációs modul C# 4.0 programozási nyelven készült Visual Studio 2010 környezetben. (Vágner et al., 2011 A), (Vágner et al., 2011 B), (Juhász et al., 2009)

Az algoritmus inputként egy 3 csatornás EKG jelet kap. Ugyancsak inputként kapja a szívütések annotációit és a típusait. Az algoritmus az annotációk segítségével feldarabolja az EKG jelet, melynek eredményeként egy szívütéshez tartozó jelsorozatot kapunk. Ezután egy szívütést egy kétdimenziós pontpárral jellemzünk, amelyet wavelet transzformáció segítségével nyerünk ki az eredeti jelből. A pontpárok halmazán a GridOPTICS klaszterező algoritmust alkalmazzuk, mivel az OPTICS túl lassú volt erre a feladatra, ezenkívül felismertük, hogy sok pontnak ugyanazok a koordinátái vagy nagyon közel vannak egymáshoz. Ez amiatt van, hogy a különböző szívütéstípusokat reprezentáló pontok szóródása nem teljesen véletlenszerű. Az algoritmus végeredményeképp a szívütések klaszterei állnak elő.

A fejezetben ezután a vizualizációs interfészt mutatom be.

A modul lehetővé teszi, hogy a felhasználó a klasztereket manuálisan további csoportokba bonthassa.

A felhasználói visszajelzések azt mutatják, hogy ez a technika hatékonyan támogatja a HOLTER EKG jelek kiértékelését.

6.2.2 Vérnyomásmérés

A 3.3. fejezet először egy rövid összefoglalót ad a különböző vérnyomásmérési módszerekről, bővebben az oszcillometriás eljárást mutatja be. Leírást ad arról, hogy az osszcillometriás vérnyomásmérés során milyen hibák történhetnek. Majd bemutatja, miért fontos a járóbeteg ellátás során a 24 óra alatt végzett többszöri (akár negyedóránként) vérnyomásmérés.

Ezután a Labtech Kft. Cardiospy rendszerének egy új modulja kerül bemutatásra, amely mikrokontroller által gyűjtött oszcillometriás vérnyomásmérések PC oldali feldolgozását valósítja meg. A mikrokontrollernek korlátozott a memória és processzorbeli kapacitása, emiatt pontatlan eredményeket ad vagy nem tud eredményt számolni. A PC-oldali alkalmazás a méréseket sokkal pontosabban tudja kiértékelni. A kardiológus vagy a kutató az alkalmazás használatával elemezni tudja a vérnyomásmérés feldolgozásának a lépéseit. Az alkalmazás segít felismerni a mérés alatt felmerülő hibákat. (Vágner et al., 2014)

Az algoritmus fő lépései a következőek:

1. Az algoritmus a felvételeket mérésekre bontja fel. Egyszerre csak egy méréssel dolgozik.
2. A mérésből létrehozza az oszcillogramot egy sávszűrő használatával.
3. Megkeresi az oszcillogramon a lokális minimumok és maximumok helyét és értékeit.
4. Létrehoz egy hisztogramot a lokális szélsőértékekből. A hisztogram a nyomásváltozást mutatja a minimum pontban.
5. Az algoritmus a hisztogramra egy burkológörbét, azaz egy polinomot illeszt.
6. Meghatározza a burkoló görbe maximumának helyét és értékét.
7. Meghatározza a vérnyomásértékeket.
8. Végül információt ad arról, hogy az eredmény elfogadható-e vagy nem.

Az alkalmazás a feldolgozás lépéseinek eredményét interaktív módon megjeleníti. Azaz, ha a felhasználó paramétert változtat, a skála, az egérpozíció, illetve a felület objektumai azonnal módosulnak. Az interaktív vizualizációs felület segíti a felhasználót a vérnyomásmérési információk jobb megértésében. Az alkalmazás elsősorban több mérést tartalmazó hosszú (akár 24 óra alatt készített) felvételek feldolgozásához készült, így könnyedén navigál egy felvétel mérései között. Egy validációs alkalmazást is készítettünk, amely a vérnyomásmérő algoritmus validálását támogatja. Az alkalmazás referenciával ellátott mérések tömegét dolgozza fel, statisztikai adatokat jelenít meg a mérésekről, illetve a Bland-Altman diagramon megmutatja a referencia adatok és az algoritmus eredményei közötti különbséget.

6.3 Adatbázis-programozás oktatása

A 4. fejezetben a "learning by doing" vagy aktív tanulási módszert mutatom be, amelyet egy adatbázis-programozás témájú tantárgy keretein belül használtam. A módszert napjainkban is sikerrel alkalmazzák. A programozás oktatásában nagyon fontos, hogy a hallgatók önállóan gyakoroljanak. Az aktív tanulási módszer rákényszeríti őket arra, hogy önállóan írjanak programkódokat és használják az adatbázis-kezelő rendszert. A tanár feladata, hogy végigvezesse a hallgatókat a tananyagon és támogassa őket. Az oktató személyre szabott visszajelzéseket ad a hallgatóknak, motiválja őket és megbeszéli velük a feladatokat és a megoldásokat. (Vágner, 2014)

Az aktív tanulási módszer támogatására egy szoftveralkalmazást használhatunk, amely segít a hallgató, a feladatok és a megoldások adminisztrálásában, ezen kívül a hallgatók teljesítményét is követhetjük vele. A programozás oktatása esetén

a szintaktikai ellenőrzésben is segíthet, és esetleg a szemantikai ellenőrzéshez is adhat támogatást. (Vágner, 2015)

Először a kapcsolódó irodalmat tekintem át, majd bemutatom az oktatott kurzust. A kurzus témája Oracle SQL és PL/SQL. A laboratóriumi gyakorlatokon használt környezetet (azaz a szoftvereket) ugyancsak itt mutatom be.

Ezután leírom, hogy hogyan szerveztem meg az órai munkát, azaz milyen, az aktív tanulási módszernek megfelelő elvek mentén építettem fel az órákat. Ezek a következőek:

1. A tanulóknak a kiadott feladatokon önállóan kell dolgozniuk.
2. Az oktató minden hallgató minden megoldására ad valamilyen visszajelzést.
3. Ha a megoldás rossz, akkor a hallgató a határidő előtt adhat be másik megoldást az adott feladatra.
4. A hallgatók alapvetően az órán oldják meg a feladatot, de ha nem fejezték be azokat, akkor otthon folytathatják a feladatmegoldást.
5. A feladatokhoz határidő van rendelve.
6. A hallgatók önállóan dolgoznak, de megbeszélhetik egymással a feladatokat.
7. A gyakorlat az előadás anyagát használja, azaz az előadás új anyagát gyakorolja a gyakorlaton. A gyakorlaton nincs új anyag.
8. Az oktatónak úgy kell szerveznie az előadást, hogy a hallgató egyedül tudjon dolgozni az új tananyaggal a gyakorlaton.

Ezután az adatbázis-programozást támogató alkalmazást mutatom be. Az alkalmazás két fő részből áll: az adatbázis-objektumok, amelyekben a feladatokat és a megoldásokat tárolhatjuk és egy C# program, amelyet az oktató arra használ, hogy visszajelzéseket adjon a diákoknak a megoldásaikra. Az alkalmazás egyik fő része a szintaktikai ellenőrző, amely nem engedi feltölteni azokat a megoldásokat, amelyek nem futnak le, így a hallgatók csak a szintaktikailag helyes megoldásokat tölthetik fel.

A 4. fejezet végén az aktív tanulási módszer alkalmazásának kiértékelését írtam le. A 3 év alatt gyűjtött tapasztalatok és a hallgatói eredmények megmutatják, hogy a laboratóriumi gyakorlati eredmények jobbak, ha az aktív tanulási módszerrel tanulnak a diákok. A tantárgyat elvégzett hallgatókat önkéntes alapon megkérdeztem a módszerről. A kérdőívekből az derült ki, hogy a diákok szerették a módszert, és örülnének, ha más tárgy keretei között is így tanulhatnának. A hallgatóknak a félév során keményen kellett tanulniuk, de a félév végére olyan tapasztalatot szereztek az adatbázis-kezelő rendszer használatában, amit még sok évig nem fognak elfelejteni.

References

Abdulla, D. (2015). EKG. [online] Heartsite.com. Available at: http://www.heartsite.com/html/ekg.html [Accessed 10 Oct. 2015].

Aboy, R. (2011). Method for blood pressure measurement from noninvasive oscillometric pressure signals. *Technical Report, Tiba Medical, Inc.*

Achtert, E., Böhm, C., and Kröger, P. (2006). DeLi-Clu: Boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking. *Advances in Knowledge Discovery and Data Mining*, 3918, pp. 119-128.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 27(2), pp. 94-105.

Alfaouri, M. and Daqrouq, K. (2008). ECG signal denoising by wavelet transform thresholding. *American Journal of Applied Sciences*, 5(3), pp. 276-281.

Alzaalan, M. E., Aldahdooh, R. T., and Ashour, W. (2012). EOPTICS "Enhancement Ordering Points to Identify the Clustering Structure". *International Journal of Computer Applications*, 40(17), pp. 1-6.

AmperorDirect, (2015). *Purpose of ECG / EKG (Electrocardiogram).* [online] Available at: https://www.amperordirect.com/pc/help-ecg-monitor/z-ekg-purpose.html [Accessed 10 Oct. 2015].

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999) OPTICS: Ordering Points to Identify the Clustering Structure. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pp. 49–60.

Ball-llovera, A., Del Rey, R., Ruso, R., Ramos, J., Batista, O., and Niubo, I. (2003). An experience in implementing the oscillometric algorithm for the noninvasive determination of human blood pressure. *Engineering in Medicine and Biology Society, Proceedings of the 25th Annual International Conference of the IEEE*, vol. 4, pp. 3173–3175.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* New York: Springer.

Bland, J. and Altman, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, pp. 307–310.

Bland, J. and Altman, D. (1999). Measuring agreement in method comparison studies. *Statistical Methods in Medical Research*, 8 (2), pp. 135–160.

BPLab User's guide, (2015). [online] Available at: http://www.strumedical.com/admin/allegati/278-manuale.pdf [Accessed 10 Oct. 2015].

Brecheisen, S., Kriegel, H., and Pfeifle, M. (2006). Multi-step density-based clustering. *Knowledge and Information Systems*, 9(3), pp. 284-308.

Breunig, M. M., Kriegel, H.-K., and Sander, J. (2000). Fast hierarchical clustering based on compressed data and OPTICS. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 232–242.

Bulletin of Software Engineering Bsc program at University of Debrecen, (2007). [online] Available at: http://www.inf.unideb.hu/oktatas/?cat=&site=hallgato/nappali/oklevel_kovetelmeny/pti_2007 English version: http://www.englishstudies.sci.unideb.hu/images/documents/SoftwareIT_BSc_2011.pdf [Accessed 11 Oct. 2015].

Cepek, M., Chudacek, V., Petrik, M., Georgoulas, G., Stylios, C., and Lhotska L. (2007). Comparison of inductive modelling method to other classification methods for Holter ECG. *International Workshop on Inductive Modelling*. pp. 229-241.

Chouhan, V. S. and Mehta, S. S. (2008). Detection of QRS complexes in 12-lead ECG using adaptive quantized threshold. *International Journal of Computer System and Network Security*, 8(1), pp. 155-163.

Christov, I. I. (2004). Real time electrocardiogram QRS detection using combined adaptive threshold. *BioMedical Engineering OnLine*, 3:28.

Ciocoiu, I. B. (2009). ECG signal compression using 2D wavelet foveation. *International Journal of Advanced Science and Technology*, 13, pp. 15-26.

Darong, H. and Peng, W. (2012). Grid-based DBSCAN algorithm with referential parameters. *Physics Procedia*, 24, pp. 1166-1170.

De Chazal, P., O'Dwyer, M., and Reilly, R.B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, 51 (7), pp. 1196-1206.

Dilmac, S. and Korurek, M. (2015). ECG heartbeat classification method based on modified ABC algorithm. *Applied Soft Computing*, 36, pp. 641-655.

Dotsinsky, I. A. and Stoyanov, T. V. (2004). Ventricular beat detection in single channel electrocardiograms. *BioMedical Engineering OnLine*, 3:3.

Drake, J. R. (2012). A critical analysis of active learning and an alternative pedagogical framework for introductory information systems courses. *Journal of Information Technology Education: Innovations in Practice*, 11, pp. 39-52.

Dufour, R., DuFour, R., Eaker, R., and Many, T. (2010). *Learning by Doing: A Handbook for Professional Communities at Work*, United States of America: Solution Tree Press

Elgendi, M., Jonkman, M., and De Boer, F. (2009). Improved QRS detection algorithm using dynamic thresholds. *International Journal of Hybrid Information Technology*, 2 (1), pp. 65-80.

Ester, M., Kriegel, J.-P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 96)*, no. 34, pp. 226–231.

Fränti, P. and Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5), pp. 761-775.

Freud, R. J. and Wilson, W. J. (2003). *Statistical Methods*, Second Edition. California: Elsevier.

Gábor, A. and Juhász, I. (2007). *PL/SQL Programozás (PL/SQL Programming)*, Budapest: Panem.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering.* Philadelphia: SIAM, Society for Industrial and Applied Mathematics.

Geddes, L. A. (1991). *Handbook of Blood Pressure Measurement.* New York: Humana Press.

Ghuman, N., Campbell, P., and White, W. B. (2009). Role of ambulatory and home blood pressure recording in clinical practice. *Current Cardiology Reports*, 11(6), pp. 414–421.

Gionis, A., Mannila, H., and Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), pp. 1-30.

Gogoulou, A., Gouli, E., and Grigoriadou, M. (2009). Teaching programming with ECLiP didactical approach, *International Conference on Cognition and Exploratory Learning in Digital Age*, pp. 204-211.

Han, J. and Kamber, M. (2006). *Data Mining.* Amsterdam: Elsevier.

Hinneburg, A. and Gabriel, H.-H. (2007). DENCLUE 2.0: Fast clustering dased on kernel density estimation. *Advances in Intelligent Data Analysis VII*, 4723, pp. 70–80.

Hinneburg, A. and Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. *Proceedings of the 4th International Conference on Knowledge Discovery and Datamining (KDD 98)*, pp. 58-65.

Hinneburg, A. and Keim, D. A. (1999). Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering. *Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 506–17.

Houghton, A. R. and Gray, D. (2008). *Making Sense of the ECG*, Third Edition. London: Arnold.

iHealth, (2015). *iHealth*. [online] Available at: http://www.ihealthlabs.com/blood-pressure-monitors/wireless-blood-pressure-monitor/ [Accessed 10 Oct. 2015].

Juhász I., Vágner A., Balázs Á., és Kuk J. (2009. december 12.). QRS template készítése hosszúidejű EKG vizsgálatokhoz. *Kardiológiai termékekben használható kiértékelő algoritmusok kutatása és fejlesztése, konferencia nap*, Debrecen.

Kärkkäinen, I. and Fränti, P. (2002). Dynamic local search algorithm for the clustering problem. *Research Report A-2002-6*, University of Joensuu.

Karypis, G., Han, E. H., and Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, *IEEE Transactions on Computers*, 32 (8), pp. 68-75.

Kobalava, Z. D., Kotovskaia, I. V., Rusakova, O. S., and Babaeva, L. A. (2003). Validation of UA-767 plus device for self-measurement of blood pressure. *Clinical Pharmacology and Therapy*, 12, pp. 70–72.

Kósa M., Pánovics J., and Gunda L. (2004). An evaluating tool for programming contests. *6th International Conference on Applied Informatics*, I, pp. 163–172.

Labtech, (2015). *Labtech Ltd. - Home*. [online] Available at: http://www.labtech.hu [Accessed 10 Oct. 2015].

Labtech, Electrode placement, (2015). [online] Available at: http://www.labtech.hu/downloads/productdoc/holter/Electrode_placement-eng.pdf [Accessed 10 Oct. 2015].

Lackovic, I. (2003). Engineering aspects of noninvasive blood pressure measurement with the emphasis on improvement of accuracy. *Medical and Hospital Engineering*, 41, pp. 73–85.

Lee, J., Kim, J., and Yoon, G. (2001). Digital envelope detector for blood pressure measurement using an oscillometric method. *Journal Medical Engineering and Technology, Proceedings of the 23rd Annual International Conference of the IEEE*, 1, pp. 126–128.

Lidl, R. and Pilz, G. (1998). *Applied Abstract Algebra*. New York: Springer.

Lin, C.-T., Liu, S.-H., Wang, J.-J., and Wen, Z.-C. (2003). Reduction of interference in oscillometric arterial blood pressure measurement using fuzzy logic. *IEEE Transactions on Biomedical Engineering*, 50(4), pp. 432–441.

Lin, H.-C. (2007). Specialised non-invasive blood pressure measurement algorithm. *Master's thesis*, Auckland University of Technology.

Ma, L., Gu, L., Qiao, S., Wang, J. (2014). G-DBSCAN: An improved DBSCAN clustering method based on grid. *Advanced Science and Technology Letters*, 74, pp. 23-28.

Ma, S., Wang, T., Tang, S., Yang, D., and Gao, J. (2003). A new fast clustering algorithm based on reference and density. *Advances in Web-Age Information Management*, 2762, pp. 214-225.

Mann, A. K. and Kaur, N. (2013). Grid density based clustering algorithm. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(6) pp. 2143-2147.

Martis, R., Acharya, U., and Min, L. (2013). ECG beat classification using PCA, LDA, ICA, and discrete wavelet transform. *Biomedical Signal Processing and Control*, 8(5), pp. 437-448.

Mason, R. T. (2013). A database practicum for teaching database administration and software development at Regis University, *Journal of Information Technology Education: Innovations in Practice*, 12, pp. 159-168.

Micó, P., Cuesta, D., and Novák, D. (2005). Clustering improvement for electrocardiographic signals. *Image Analysis and Processing*, 3617, pp. 892-899.

Moore, M., Binkerd, C., and Fant, S. (2002). Teaching web-based database application development: an inexpensive approach, *United States of America: Journal of Computing Sciences in Colleges*, 17(4), pp. 58-63.

Myers, M. G. (2010). A proposed algorithm for diagnosing hypertension using automated office blood pressure measurement. *Journal of Hypertension*, 28(4), pp. 703-708.

O'Brien, E., Petrie, J., Littler, W., de Swiet, M., Padfield, P. L., Altman, D. G., Bland, M., Coats, A., and Atkins, N. (1993). The British hypertension society protocol for the evaluation of blood pressure measuring devices. *Journal of Hypertension*, 11, pp. 43–62.

O'Brien, E., Atkins, N., Stergiou, G., Karpettas, N., Parati, G., Asmar, R., Imai, Y., Wang, J., Mengden, T., and Shennan, A. (2010). European Society of Hypertension International Protocol revision 2010 for the validation of blood pressure measuring devices in adults. *Blood Pressure Monitoring*, 15(1), pp. 23-38.

Oracle Documentation, (2013). *Oracle Database Online Documentation 11g Release 2 (11.2).* [online] Available at: http://www.oracle.com/pls/db112/homepage [Accessed 10 Oct. 2015].

Pan, J. and Tompkins, W. (1985). A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, 3, pp. 230-236.

Parikh, M. and Varma, T. (2014). Survey on different grid based clustering algorithm. *International Journal of Advance Research in Computer Science and Management Studies*, 2(2), pp. 427-430.

Patwary, M. A., Palsetia, D., Agrawal, A., Liao, W. Manne, F., and Choudhary, A. (2013) Scalable parallel OPTICS data clustering using graph algorithmic techniques. *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 49, pp. 1–12.

Polya, G. (1957). *How to Solve It*, United States of America: Princeton University Press.

Ramakrishna, M. V. (2000). A learning by doing model for teaching advanced databases. *Proceedings of the Australasian conference on Computing education*, pp. 203-207.

Rani, R., Chouhan, V. S., and Sinha, H. P. (2015). Automated detection of QRS complex in ECG signal using wavelet transform. *International Journal of Computer Science and Network Security*, 15(1), pp. 1-5.

Roberts, J. (2011). *Beyond learning by doing.* New York: Routledge.

Sander, J., Qin, X., Lu, Z., Niu, N., and Kovarsky, A. (2003). Automatic extraction of clusters from hierarchical clustering representations. *Proceedings of the 7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 75-87.

Sapiński, A. (1992). Standard algorithm of blood-pressure measurement by the oscillometric method. *Medical & Biological Engineering & Computing*, 30(6), pp. 671-671.

Sapiński, A. (1997). Theoretical basis for proposed standard algorithm of blood pressure measurement by the sphygmooscillographic method. *Journal of Clinical Engineering*, 22(3), pp. 171-174.

Schank, R. G., Berman T. R., and Macpherson, K. A. (1999). Learning by doing, I*nstructional Design Theories and Models, United States of America: Lawrence Erlbaum Associates*, pp. 161-182.

Schneider, J. and Vlachos, M. (2013). Fast parameterless density-based clustering via random projections. *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, pp. 861–866.

Shahram, M. and Nayebi, K. (2001). Classification of multichannel ECG signals using a cross-distance analysis. *Engineering in Medicine and Biology Society. Proceedings of the 23rd Annual International Conference of the IEEE*, 3, pp. 2182-2185.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. (2000). WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal The International Journal on Very Large Data Bases*, 8(3-4), pp. 289-304.

Silabs, (2015). [online] Available at: https://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F064-short.pdf [Accessed 10 Oct. 2015]

Skinner, B. (1968). *The Technology of Teaching.* New York: Meredith Corporation.

Sörnmo, L. and Laguna, P. (2005). *Bioelectrical Signal Processing in Cardiac and Neurological Applications.* Amsterdam: Elsevier Academic Press.

Sorvoja, H. (2006). *Noninvasive Blood Pressure Pulse Detection and Blood Pressure Determination.* Oulu: University of Oulu.

Tan, P., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining.* Boston: Pearson Addison Wesley.

Tsipouras, M. G., Voglis, C., Lagaris, I. E., and Fotiadis, D. I. (2005). Cardiac arrhythmia classification using support vector machines. *The 3rd European Medical and Biological Engineering Conference.*

Vágner, A. (2014). Let's learn database programming in an active way, *Teaching Mathematics and Computer Science*, 12(2), pp. 213–228.

Vágner, A. (2015). Software application for supporting the education of database systems. *Acta Didactica Napocensia*, 8(3), pp. 23-28.

Vágner, A. (in press). The GridOPTICS clustering algorithm. *Intelligent Data Analysis*, 20(5).

Vágner, A., Farkas, L., and Juhász, I. (2011 A). Clustering and visualization of ECG signals, *Third International Conference on Software, Services and Semantic Technologies - S3T 2011, Advances in Inteligent and Soft Computing*, 101, pp. 47-51.

Vágner, A., Juhász, I., Kuk, J., and Balázs, Á. (2011 B). Clustering of ECG signals. *International Conference on Applied Informatics*, Eger, II, pp. 129-137.

Vágner, A., Vámosi B., and Juhász, I. (2014). Visualization and off-line processing of blood pressure signals. *Proceedings of the International Conference on Health Informatics*, pp. 393-398.

Wang, J.-J., Lin, C.-T., Liu, S.-H., and Wen, Z.-C. (2002). Model-based synthetic fuzzy logic controller for indirect blood pressure measurement. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics archive, 32(3), pp. 306–315.

Welch Allyn, (2015). [online] Available at: http://intl.welchallyn.com/documents/Blood%20Pressure%20Management/7171WAWorkbook_AUG5.pdf [Accessed 10 Oct. 2015].

Ye, C., Kumar, B., and Coimbra, M. (2012). Heartbeat classification using morphological and dynamic features of ECG signals. *IEEE Transactions on Biomedical Engineering*, 59(10), pp. 2930-2941.

Yue, S., Wei, M., Li, Y., and Wang, X. (2007). Ordering grids to identify the clustering structure. *Advances in Neural Networks – ISNN 2007*, 4492, pp. 612-619.

Zhang, T., Ramakrishnan, R., and Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2), pp. 141-182.

Zhao, Y., Cao, J., Zhang, C., and Zhang, S. (2011). Enhancing grid-density based clustering for high dimensional data. *Journal of Systems and Software*, 84(9), pp. 1524-1539.

Zheng, D., Giovannini, R., and Murray, A. (2011). Effect of talking on mean arterial blood pressure: Agreement between manual auscultatory and automatic oscillometric techniques. *In Computing in Cardiology*, 38, pp. 841–844.

Zheng, G., Tang, T., and Huang, Y. (2007). Automatic selection of dynamic electrocardiogram waveform by knowledge discovery technologies. *International Conference on Business and Information.*

Zidelmal, Z., Amirou, A., Ould-Abdeslam, D., and Merckle, J. (2013). ECG beat classification using a cost sensitive classifier. *Computer Methods and Programs in Biomedicine*, 111(3), pp. 570-577.

List of publications

Peer-reviewed papers

Geda, G. and Vágner, A. (2006). Solving ordinary differential equation systems by approximation in a graphical way, *Annales Mathematicae et Informaticae*, 33, pp.: 57-68.

Vágner, A., Farkas, L., and Juhász, I. (2011). Clustering and visualization of ECG signals, *Third International Conference on Software, Services and Semantic Technologies - S3T 2011, Advances in Inteligent and Soft Computing*, 101, pp. 47-51.

Vágner, A., Vámosi B., and Juhász, I. (2014). Visualization and off-line processing of blood pressure signals. *Proceedings of the International Conference on Health Informatics*, pp. 393-398.

Szabo, R., Farkas, K., Ispany, M., Benczur, A. A., Batfai, N., Jeszenszky, P., Laki, S., Vagner, A., Kollar, L., Sidlo, C., Besenczi, R., Smajda, M., Kover, G., Szincsak, T., Kadek, T., Kosa, M., Adamko, A., Lendak, I., Wiandt, B., Tomas, T., Nagy, A. Z., and Feher, G. (2013). Framework for smart city applications based on participatory sensing. *Cognitive Infocommunications, IEEE 4th International Conference on*, pp.295-300.

Vágner, A. (2014). Let's learn database programming in an active way, *Teaching Mathematics and Computer Science*, 12(2), pp. 213–228

Szincsák, T. and Vágner, A. (2014). Data structure to store GTFS data efficiently on mobile devices. *Journal of Computer Science and Software Application*, 1 (1), pp.27-41.

Vágner, A. and Zsakó, L. (2015). Negative effects of learning spreadsheet management on learning database management. *Acta Didactica Napocensia*, 8(2), pp. 1-6.

Vágner, A. (2015). Software application for supporting the education of database systems. *Acta Didactica Napocensia*, 8(3), pp. 23-28.

Vágner, A. (in press). The GridOPTICS clustering algorithm. *Intelligent Data Analysis*, 20(5).

Conference proceedings

Vágner, A., Juhász, I., Kuk, J., and Balázs, Á. (2011). Clustering of ECG signals. *International Conference on Applied Informatics*, Eger, II, pp. 129-137.

Juhász, I., Kósa, M., and Vágner, A. (2011). Teaching satabase systems at the Faculty of Informatics at the University of Debrecen. *International Conference on Applied Informatics*, Eger, II, pp. 9-15.

Szincsák, T. and Vágner, A. (2014). Public transit schedule and route planner application for mobile devices. *International Conference on Applied Informatics*, Eger.

Vágner A. (2008). Adatbázisrendszerek oktatása az Eszterházy Károly Főiskolán. *Informatika a Felsőoktatásban*, Debrecen. A44.

Vágner A. (2011). Haladó adatbázis ismeretek oktatása a Debreceni Egyetem Informatikai Karán, *Informatika a Felsőoktatásban*, Eger, pp: 399-405.

Conference talks

Adamkó, A., Kádek, T., Kollár, L., Kósa, M., Szincsák, T., and Vágner, A. (29 February, 2014). From university calendars to smart university administration. *International Conference on Applied Informatics*, Eger.

Adamkó, A., Kádek, T., Kollár, L., Kósa, M., Szincsák, T., and Vágner, A. (29 February, 2014). Crowdsourcing-based evaluation of seminar exercises and test case development. *International Conference on Applied Informatics*, Eger.

Juhász I., Vágner A., Balázs Á., és Kuk J. (2009. december 12.). QRS template készítése hosszúidejű EKG vizsgálatokhoz. *Kardiológiai termékekben használható kiértékelő algoritmusok kutatása és fejlesztése, konferencia nap*, Debrecen.

Vágner A. (2013. január 25-27). A PL/SQL programozási nyelv szoftverrel támogatott oktatása. *Matematika és Informatika Didaktikai Konferencia* (MIDK 2013), Nagyvárad.

Vágner A. és Szincsák T. (2014. November 6-7). Mobil eszközök szenzorai által gyűjtött adatok felhasználási lehetőségei. *International Conference on Future RFID Technologies and host the Workshop on Smart Applications for Smart Cities*, Eger.

Books

Gábor A., Gunda L., Juhász I., Kollár L., és Vágner A. (2003). *Az Oracle és a Web. Haladó Oracle9i Ismeretek*, Budapest: Panem.

Lecture notes

Vágner A. és Juhász I. (2011). Adatbázis-adminisztráció, elektronikus jegyzet, Debrecen.

Vágner A. (2015). Bevezetés az ABAP programozásba, Debrecen.

Vágner A. (2015). Introduction into ABAP programming, Debrecen.

Book translations

Loney, K. és Koch, G. (2001). *ORACLE8i - Teljes Referencia*, (szerkesztő: Juhász I., fordítók: Juhász I., Csordás A., Fajta R., és Vágner A.), Budapest: Panem.

Abbey, M., Corey, M. J., és Abramson, I. (2001). *ORACLE8i - Kézikönyv Kezdőknek*, (szerkesztő: Juhász I., fordítók: Csordás A., Fazekas R., Juhász I., Kókai F., Mohai G., Rákosi P., és Vágner A.), Budapest: Panem.

Sebesta, R. W. (2005). *A World Wide Web Programozása*, (fordítók: Altfatter Z., Borsi É. E., Gábor Zs., Juhász I., és Vágner A.), Budapest: Panem.

Loney, K. (2006). *ORACLE Database 10g Teljes Referencia* (fordítók: Agócs L., Altfatter Z., Borsi É. E., Juhász I., Mohai G., Rákosi P., Román Á., és Vágner A.), Budapest: Panem.