applied sciences

MDPI

Article

Multi-Variable, Multi-Layer Graphical Knowledge Unit for Storing and Representing Density Clusters of Multi-Dimensional Big Data

K. K. L. B. Adikaram [1,2,3,*], Mohamed A. Hussein [1,†], Mathias Effenberger [2,†] and Thomas Becker [4,†]

1. Group Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, Freising 85354, Germany; hussein@wzw.tum.de
2. Bavarian State Research Center for Agriculture, Institute for Agricultural Engineering and Animal Husbandry, Vöttinger Straße 36, Freising 85354, Germany; mathias.effenberger@lfl.bayern.de
3. Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, Kamburupitiy 81100, Sri Lanka
4. Lehrstuhl für Brau- und Getränketechnologie, Technische Universität München, Weihenstephaner Steig 20, Freising 85354, Germany; tb@bgt.wzw.tum.de

* Correspondence: lasantha@daad-alumni.de; Tel.: +94-412292200

† These authors contributed equally to this work.

Academic Editor: Antonio Fernández-Caballero

Received: 25 October 2015; Accepted: 15 March 2016; Published: 5 April 2016

Abstract: A multi-variable visualization technique on a 2D bitmap for big data is introduced. If A and B are two data points that are represented using two similar shapes with m pixels, where each shape is colored with RGB color of $(0, 0, k)$, when $A \cap B \neq \phi$, adding the color of $A \cap B$ gives higher color as $(0, 0, 2k)$ and the highlight as a high density cluster, where RGB stands for Red, Green, Blue and k is the blue color. This is the hypothesis behind the single variable graphical knowledge unit (GKU), which uses the entire bit range of a pixel for a single variable. Instead, the available bit range of a pixel is split, and a pixel can be used for representing multiple variables (multi-variables). However, this will limit the bit block for single variables and limit the amount of overlapping. Using the same size k (>1) bitmaps (multi-layers) will increase the number of bits per variable (BPV), where each (x, y) of an individual layer represents the same data point. Then, one pixel in a four-layer GKU is capable of showing more than four billion overlapping ones when BPV = 8 bits ($2^{(BPV \times \textit{number of layers})}$) Then, the 32-bit pixel format allows the representation of a maximum of up to four dependent variables against one independent variable. Then, a four-layer GKU of w width and h height has the capacity of representing a maximum of $(2^{(BPV \times \textit{number of layers})}) \times m \times w \times h$ overlapping occurrences.

Keywords: knowledge representation; continuous learning; cluster identification; big data

1. Introduction

Multi-variable analysis and graphical representation of data are two demanding factors in the field of data analysis. Multi-variable analysis is a method for depicting the correlation between variables, while graphical representation of data is a very efficient tool for abstracting information in a multi-variable dataset. In addition, graphical representation usually conveys the intended information more easily than text or numerical values [1]. However, when the numbers of available data are high, it is difficult to represent all of the data points of a scattered dataset as a plot due to the high number of overlapping data points, also called occlusion or over-plotting. This is one of the main issues in the field of data visualization, which leads to the loss of data in projection [2]. On the other hand, when the numbers of variables are high, special techniques are required to represent multi-dimensions on a two-dimensional or three-dimensional space. The biggest challenge is to visualize multi-dimensional big data.

Density cluster identification is a technique that is used in the field of knowledge discovery in databases (KDD) [3–6]. Furthermore, density cluster identification is a tool that is used to identify the correlation between variables. In cluster visualization, first, relevant density clusters were identified by means of a suitable algorithm, and then, those identified clusters were visualized by means of a suitable data visualization technique. Therefore, data visualization is usually a representation method, but not a cluster identification method [7]. The real cluster identification is done by algorithms. Of course, good visualization techniques allow viewers to identify clusters easily. However, this does not imply that these data visualizations are made for identifying clusters. As in multi-dimensional data analysis, cluster visualization also suffers when the numbers of data points are high, especially due to overlapping.

The term "big data" refers to datasets for which the size is beyond the capabilities of current database technology [6,8,9]. According to Karimi, big data is a "collection of databases so large or complex that it becomes difficult to process using regular database management tools or traditional data processing applications" [10]. In this work, big datasets are addressed from the perspectives of processing and representation. In data visualization techniques, it is hard to identify a specific method that is capable of visualizing big data [6]. The major and most practical reason for this is that there is no adequate space in a plot to visualize all of the data points. In the domain of a scattered dataset, overlapping data are another barrier for visualizing big data. On the other hand, according to the definition, big data are data requiring extreme methodologies for processing, as well as storage [11].

2. Related Work

Scatterplot matrices [12], parallel coordinates [13], Andrews' curves [14,15], Radviz [16] and star coordinates [17] are the most popular techniques used for visualization of multi-dimensional data. Except for star coordinates, all of the other mentioned methods display the multi-dimensional data on a lower-dimensional space in a way that additional effort is needed for understanding the original number of dimensions [17]. In contrast, star coordinates directly visualize dimensions as groups in the form of high density clusters in a two-dimensional plot [17]. Overlapping and a higher number of scatter plots in a scatterplot matrix convey poor visualization [18]. Parallel coordinates are not suitable for visualizing a higher number of records [2]. When there are higher numbers of overlapping points, Radviz does not provide the expected output [2,17]. Andrews' curves do not support visualizing clusters of multidimensional space even though they support a large number of data points [2] and require higher computational time for generating curves [19]. The star coordinates suffer from overlapping of clusters of variables [17,20]. These facts imply that all of the above-mentioned techniques are facing the lack of capabilities for visualizing multi-dimensional big data, in most cases due to overlapping data points.

In reality, it is normal to observe missing data [21,22] and out of range data or outliers [23,24] in most data acquisition systems. It is not always necessary to impute missing and out of range data. However, it is good to visualize the amount of missing and out of range data, to understand the magnitude and the nature of the distribution of such data. In the domain of big data, it is very important to know the quantity and the nature of such data for better comprehension of the quality of the data. When dealing with big data, missing and out of range data identification requires considerable computational effort. Nevertheless, in all of the mentioned methods, there is no standard mechanism included for showing missing and out of range data.

3. Concept of the Graphical Knowledge Unit

In 2015, a new way of representing density clusters was introduced for one dependent and one independent variable using a bitmap. The method was named the "graphical knowledge unit" (GKU) [25]. Overlapping data points are the main features that are used for creating a GKU. A higher number of overlapping ones makes a GKU more meaningful and understandable. From its origin, a GKU does not suffer from overlapping data failures. Mainly, a GKU visualizes data in the form of density clusters. Mostly, it is possible to identify high density visually and by means of an algorithm, depending on the color density. In addition, a GKU provides a mechanism for visualizing the amount and nature of the distribution of both missing and out of range data points along with the density clusters. Thus, the GKU is sought as a reliable approach for big data visualization, especially overlapping types.

Nevertheless, the GKU is not a solution for visualizing multi-variable environments. An extended usage of the GKU for representing multi-variables using the same concept of the overlapping of data points is investigated in this work. As in the GKU, the success of multi-variable multi-layer (MVML)-GKU depends on having high data density for clustering purposes. Figure 1a shows a GKU used for representing 35,864 data points, and Figure 1b, c show the 3D representation and heat map representation of the GKU. Usually, for plotting 3D maps or heat maps, it is necessary to process the data for creating the required matrix. In contrast, while using a GKU, it is not necessary to create such a matrix and when adding data points. A GKU gradually forms the density areas by incrementing the color intensity in overlapped regions.

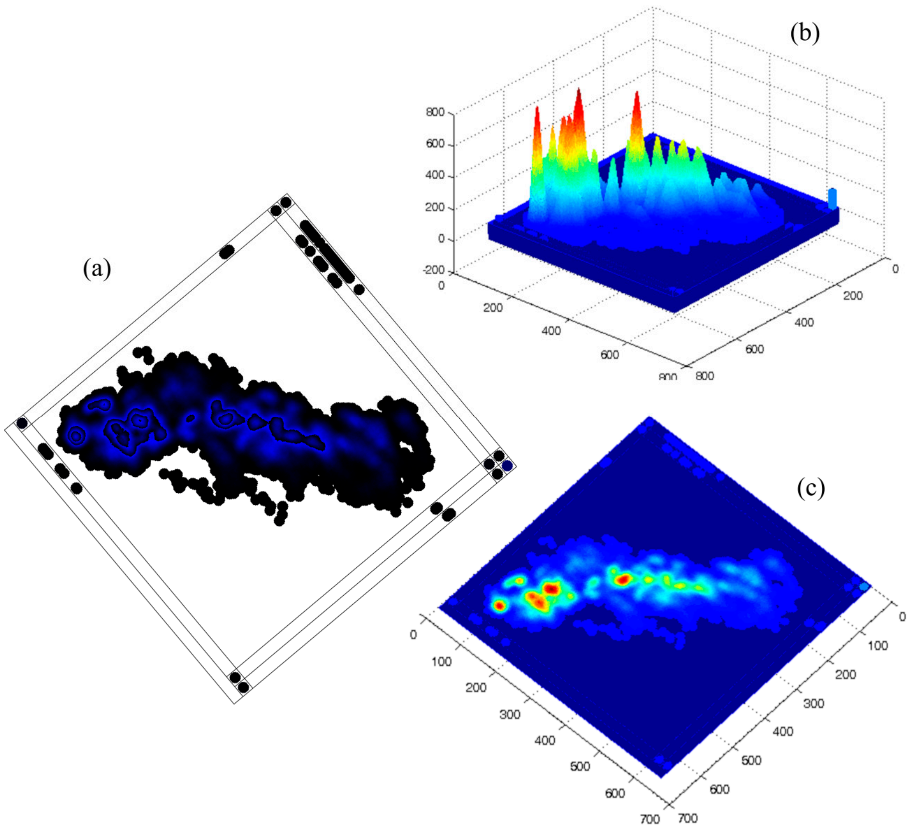

Figure 1. (**a**) Representation of 35,864 data points a graphical knowledge unit (GKU); (**b**) representation of the processed data of the same data in 3D; (**c**) representation of the processed data of the same data in a heat map.

In the proposed method, the concept of the GKU is used to represent a multi-variable big data environment in 2D space. Furthermore, the proposed method generates density clusters of all variables on the same bitmap, which can be identified by the naked eye. Most importantly, the proposed method does not suffer from overlapping data points and requires more overlapping for better visualization of density clusters. Thus, the proposed method is a robust density cluster representation and data visualization technique for multi-variable big data. Due to the very flexible nature of the proposed method, it will help to represent big data captured from dynamic bioprocesses.

4. Methodology

A GKU with a 24-bit bitmap, the RGB pixel format, is used for representing data points. Furthermore, it is possible to use a 32-bit bitmap with the alpha-RGB (ARGB) pixel format for representing the GKU. In both situations, a maximum of 24 bits or 32 bits can be allocated for representing a single variable against another variable.

Instead of allocating the whole bit range of a pixel to a single variable, it is possible to divide the bits among several variables. Figure 2 shows an example of the equal allocation of bits of a 32-bit bitmap with the ARGB pixel format for four variables. Furthermore, unequal allocation is possible depending on the requirements. Here, the equal allocation of bits for the variables was mainly considered. Figure 2 elaborates on the allocation of alpha, red, green and blue portions for representing four variables, where eight bits per variable (BPV) are used. However, when the numbers of variables are high, BPV is low.

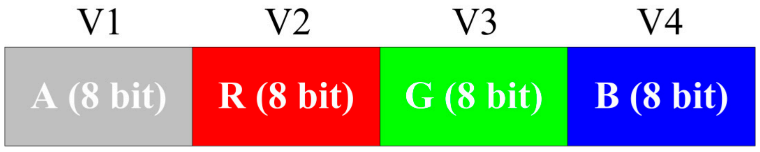

Figure 2. A single pixel of a 32-bit RGB format is split into four equal portions for representing four variables (V1, V2, V3 and V4). A, R, G and B represents the alpha, red, green and blue portions of the pixel, respectively.

When the BPV is low, it leads to fewer numbers of overlapping data points. If BPV = 8 bits, it supports the representation of the maximum of 256 overlaps and will not allow the representation of a higher number of data points. This will discourage the usage of multiple variables in the GKU. As a solution, a multi-layer bitmap GKU is proposed, which consists of several equal-sized bitmaps (Figure 3). In the GKU, when one block is full (e.g., blue), it is possible to use the adjacent block (green) to represent overlapping [25]. In contrast, in a multi-layer GKU, when an allocated block is full, it uses the relevant pixel block of another bitmap of the same size (Figure 3), which is named a layer. If the number of layers is L and the number of bits allocated for a variable is k, this technique provides $k \times L$ bits for one variable. In the GKU, the whole bit range of a pixel is allocated for one variable, which can be considered as a horizontal array. In contrast, in the MVML-GKU concept, a single variable is a vertical array with $k \times L$ bits (Figure 3). Depending on the requirement, it is possible to decide the number of layers in the MVML-GKU.

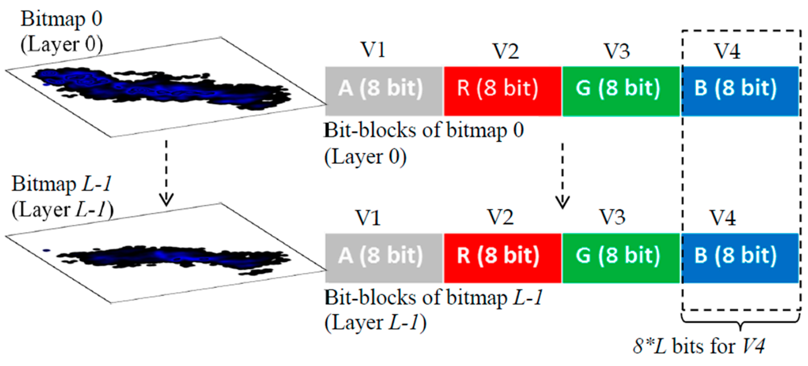

Figure 3. Four variables (V1, V2, V3 and V4) are represented using multiple bitmaps of a 32-bit RGB pixel format for forming a multi-variable multi-layer graphical knowledge unit (MVML-GKU). Each pixel is divided into four equal portions, where each portion consists of eight bits. When the 32-bit RGB pixel format is divided into four equal parts, each variable represents the alpha, red, green and blue sections of the pixel. This provides a vertical array of $k \times L$ bits for one variable.

Furthermore, each layer is numbered starting from 0, and the "place-value" is assigned for each layer according to the layer number. This assigns the same "place-value" for each bit in the relevant layer. The base of the "place-value" is in relation to the number of assigned bits for one variable, as shown in Equation (1).

$$PV^l = \left(2^{(q-p+1)} - 1\right)^l \tag{1}$$

where PV^l is the positional value of a bit in the layer l, p is the index of the starting bit of the bit block and q is the index of the ending bit of the bit block; thus, $p = 0, 1, 2, \ldots$ and $q = 0, 1, 2, \ldots$

If the $c^{(i,j)|l}_{[p,q]|l}$ represents the color value of the variable, it corresponds to bit block $[p, q]$ of the pixel (i, j) of the layer l (Figure 4). Therefore:

$$c^{(i,j)|l}_{[p,q]|l} = \sum_{k=q}^{p} b^{(i,j)|l}_{k} \times 2^{q-k} \tag{2}$$

where $b^{(i,j)|l}_{k}$ is the bit value (0 or 1) of the bit k of the bit block of a variable of pixel (i, j) of the layer l.

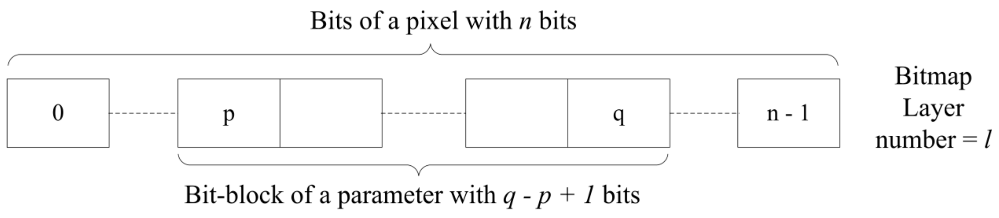
Bits of a pixel with *n* bits
0
p
q
n - 1
Bitmap Layer number = *l*
Bit-block of a parameter with *q* - *p* + *1* bits

Figure 4. Bit numbering method of a pixel in a layer.

If the total color value in relation to a certain variable corresponding to the bit block $[p, q]$ is $C^{(i,j)}_{[p,q]}$, then:

$$C^{(i,j)}_{[p,q]} = \sum_{l=0}^{L-1} c^{(i,j)|l}_{[p,q]|l} \times PV^l \tag{3}$$

Substituting from Equation (1):

$$C^{(i,j)}_{[p,q]} = \sum_{l=0}^{L-1} c^{(i,j)|l}_{[p,q]|l} \times \left(2^{(q-p+1)} - 1\right)^l \tag{4}$$

We reserved the last number of the bit block for representing the initial color of the bitmap. Therefore, we used $(2^{(q-p+1)} - 1)^l$ instead of using $(2^{(q-p+1)})^l$ as the place-value of a certain layer. For example, in the case of equal bit block allocation, if the BPV are eight, we used 255^l as the place-value of the layer l instead of 256^l. This will reserve the last value "255" for the initial color of the bit block. Whenever the whole pixel is considered, "white" is the initial color of the pixel and the bitmap.

In the concept of the GKU, one data point is represented by a shape that consists of more than one pixel (e.g., a circle) [25] as a data point shape (DPS). While adding a data point, the existing color value of all of the pixels that are overlapped by the new data point need to be updated by adding the color of the newly-added data point. In a multi-variable environment, separate data points are used for each variable. It is possible to use different DPSs for different variables. However, in this paper, the usage of the same DPS (e.g., circle) for all of the variables is discussed. In every new data addition, the color increment of the intersected area needs to be updated with a previously decided value and named the "color increment" (CI) [25]. When there is more than one variable, it is possible to use the same or different CI values for variables. In all of the DPSs, the bit block that is assigned for a certain variable is set to its initial color value while keeping all of the bits that do not belong to the considered bit block at value 1. For example, consider the situation of representing data points of four variables (V1, V2, V3 and V4) on the MVML-GKU by means of circles (radius = $r > 0$), which has a 32-bit RGB pixel format. Then, in the equal bit allocation, each variable is assigned eight bits as alpha, red, green and blue, respectively. Four separate circles are used to represent V1, V2, V3 and V4 and are colored using different "shape colors" (SC) as (CI, 255, 255, 255), (255, CI, 255, 255), (255, 255, CI, 255) and (255, 255, 255, CI), respectively. Furthermore, using (CI, 0, 0, 0), (0, CI, 0, 0), (0, 0, CI, 0) and (0, 0, 0, CI) is another possible way of implementing the four different variables. If bit block $[p, q]$ represents the CI of variable P, then P can be represented as $P_{[p,q]}$. Then, $P_{[p,q]}^{(\bar{i},\bar{j})}$ represents the pixel color of pixel $(\bar{i}, \bar{j})$ of the DPS, where $(\bar{i}, \bar{j})$ is the pixel of the DPS, which coincides with the pixel (i, j) of the MVML-GKU.

When updating the MVML-GKU, first, the $C_{[p,q]}^{(i,j)}$ is calculated using Equation (4), and then, $P_{[p,q]}^{(\bar{i},\bar{j})}$ is added. Therefore:

$$C_{[p,q]}^{(i,j)} = C_{[p,q]}^{(i,j)} + P_{[p,q]}^{(\bar{i},\bar{j})} \tag{5}$$

Finally, the value $C_{[p,q]}^{(i,j)}$ is used to update the layers of the MVML-GKU using Equation (6).

$$c_{[p,q]|l}^{(i,j)|l} = \text{QUOTIENT}\left(C_{[p,q]}^{(i,j)} - \sum_{m=l+1}^{L-1} c_{[p,q]|m}^{(i,j)|m}, \left(2^{(q-p+1)} - 1\right)^{l}\right) \tag{6}$$

where $\sum_{m=l+1}^{L-1} c_{[p,q]|m}^{(i,j)|m} = 0$ when $m > L$

When using Equation (6), it updates the MVML-GKU starting from the last layer and ends with the first layer (layer 0). However, depending on the requirements, it is also possible to update the MVML-GKU starting from Layer 0 with different approaches.

Variables with different scales or different value ranges require a larger bitmap for representing the actual value ranges of the variable values. This is not peaceable, and it is necessary to transform all of the values of different variables to one scale. To accomplish the transformation, a normalization technique known as "min-max normalization" is used. Min-max normalization performs a linear transformation on the original data mapped into a new range using Equation (7) [26,27].

$$v' = ((v - min_A)/(max_A - min_A)) \times (new\_max_A - new\_min_A) + new\_min_A \tag{7}$$

where v' is the transformed value of v, min_A is the minima among all of the data points, max_A is the maxima among all of the data points, new_max_A is the ceiling value of the new range and new_min_A is the floor value of the new range.

If the pixel at coordinates (0, 0) of a bitmap is the origin, then $new\_min_A = 0$. Then, Equation (7) becomes:

$$v' = ((v - min_A)/(max_A - min_A)) \times new\_max_A \tag{8}$$

The width of the bitmap is used for representing the independent variable, while the height of the bitmap is used for representing multiple dependent variables. Thus, the width of the bitmap is proportional to the range of independent variables, and the height is proportional to the maximum range among the ranges of all dependent variables. If the height and the width of the bitmap is h and w, respectively, then:

$$h = R_{maxD} \times F_{maxD} \tag{9}$$

$$w = R_I \times F_I \tag{10}$$

where R_{maxD} is the maximum range among the ranges of all dependent variables, R_I is the range of the independent variable, F_{maxD} is the scaling factor of the dependent variable that has the maximum range among all ranges of all dependent variables and F_I is the scaling factor of the independent variable.

In the concept of the GKU, it always uses the integer part of a number for mapping on the GKU. Because of that, it is a must to scale up data that have a small range and decimal values to minimize the influence due to truncation. Selecting appropriate values for new_max_A for Equation (8) will establish the new set of transformed data. The method of selecting a suitable value for new_max_A is mentioned in the next paragraph. Furthermore, min-max normalization transforms the data series into a positive series, which is another requirement of the GKU. These transformation data are saved in the "GKU specific data area" as a color [25].

With multiple variables, there are two possible ways of scaling. The first method is to scale all of the parameters using a single scaling factor (F). This will change all of the values of the variables while keeping the original ratio between variable values. The name "absolute scaling" will refer to such a scaling technique, from now onwards. In "absolute scaling", new_max_A for all dependent variables is not the same, but F_{iD} is the same for all dependent variables, where F_{iD} is the scaling factor of i-th dependent variable. The scaling factor of the variable with the height range can be considered as the common scaling factor for all of the dependent variables. Define R_{iD} as the range of the i-th dependent variable, and new_max_{Ai} is the new maximum of the i-th dependent variable. Then,

$$F_{iD} = F_{maxD} \tag{11}$$

From Equation (9):

$$F_{iD} = h/R_{maxD} \tag{12}$$

$$new\_max_{Ai} = R_{iD} \times F_{iD} \tag{13}$$

The second method is to scale all of the variables in a manner such that the maximum and minimum of all of the variables coincide with each other. The name "relative scaling" will refer to such scaling techniques, from now onwards. In "relative scaling", new_max_A for all dependent variables is the same, but F_{iD} is not the same for all dependent variables. "h", which is the previously decided height of the image, can be considered as the common value for new_max_{Ai}. Thus, for "relative scaling":

$$new\_max_{Ai} = h \tag{14}$$

Depending on the scaling method, a compatible new_max_A can be calculated using either Equation (13) or Equation (14).

To have a visually meaningful GKU, it is necessary to have a higher number of overlapping clusters. If there are adequate numbers of overlapping clusters, the GKU shows different density areas separated by automatically-generated "contour lines". If the number of data points is smaller, it is still possible to have more overlapping using two techniques: (1) use a bigger shape as the DPS and (2) use a higher number as the CI in SC. Using either technique, it is possible to generate higher color values in pixels and to create visible clusters. However, these techniques are important only for better visualization of the clusters. The absence of visual clusters has no effect on understanding the content of the GKU.

5. Datasets

Two online datasets are used, which are recorded from NIR spectroscopy at a biogas plant. The first dataset is recorded over a period of nearly 25 days with a frequency of 115 values per day, giving a total of 2885 data points. The second dataset is recorded over a period of 75 days with a frequency of 20 values per hour, giving a total of 21,591 data points. In both datasets, the concentration of volatile solids (VS) was selected as the independent variable, and the concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA) were selected as the dependent variables. The selected data were part of a dataset that was used to develop a near-infrared (NIR) spectroscopy online calibration for monitoring volatile solids (VS) and volatile fatty acids (VFA) as process indicators during anaerobic digestion [28].

We omitted the alpha portion of the pixel, because it is not visible. Therefore, we used three dependent variables against the independent variable. A circle of a radius of 10 pixels is selected as the DPS for three dependent variables. Three different colors were used as the CI for DPSs. All algorithms in relation to the MVML-GKU were programmed and implemented in the validation process.

6. Results and Discussion

The results included in this section represent different forms of the MLMV-GKU representations, which were obtained using different techniques. Furthermore, this section deliberates the methods and interpretation of the MLMV-GKUs. Figure 5 shows a conventional scatter plot, which presents the 2885 occurrences of three dependent variables (TVFA, AA and PA) against the independent variable (VS). Effective value ranges of TVFA, AA and PA are [0, 8], [0, 5] and [0, 5], respectively. The plot itself does not convey strong evidence on data clustering or data density. On the other hand, due to overlapping, it is difficult to identify some of the data points that may lie under other data points.

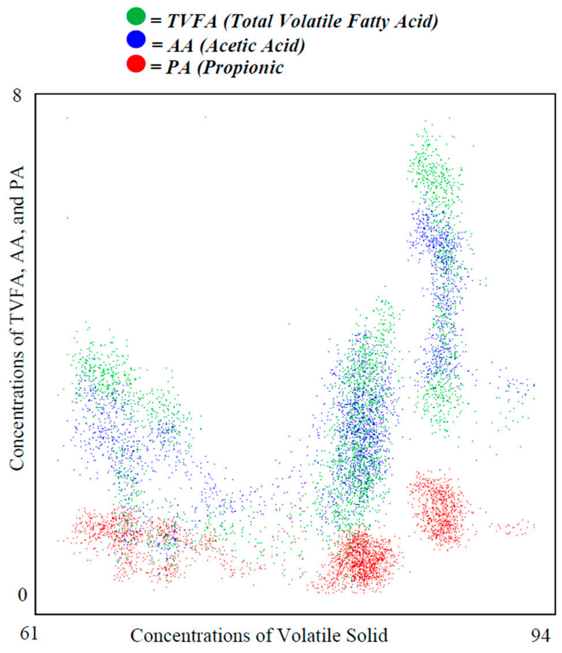

Figure 5. A conventional scatter plot for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids. Each dependent variable consists of 2885 data points. The ranges of TVFA, AA and PA are [0, 8], [0, 5], and [0, 5], respectively. Due to over-plotting, the large number of data points and the multi-variable environment, this plot itself does not provide direct useful information about data clusters. This is the major drawback of using a scatter plot for representing multi-variable big data.

Plots in Figure 6 show the plotting of the same dataset shown in Figure 7, on the MLMV-GKU of two layers using "relative scaling". The MLMV-GKU was implemented using a circle with a ten-pixel radius DPS and CI = 20. Three different RGB colors, (255, 255, 20), (255, 20, 255) and (20, 255, 255), were used for representing TVFA, AA and PA, respectively. The effective width and height of the bitmap is 400 and 400 pixels, respectively. Since "relative scaling" is used for scaling the values of variables, according to the Equation (14), new_max_A for each dependent variable is 400 pixels. Based on this, all existing ranges of all of the variables ([0, 8], [0, 5] and [0, 5]) were transformed. In Layer 0 of Figure 6, the formation of clusters that belong to different variables can be identified with the naked eye. This is one advantage of using the concept of the GKU, because it creates visibly dense clusters when there is an adequate amount of overlapping.

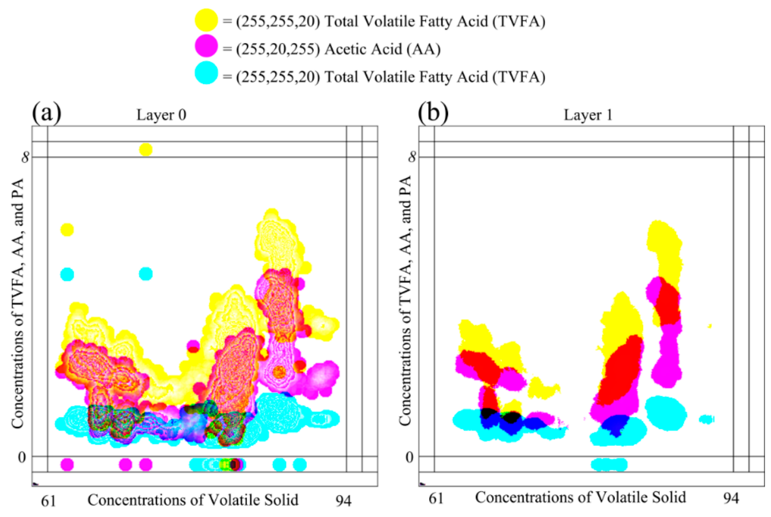

Figure 6. The MLMV-GKU of two layers. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. The effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method and used 400, 250 and 250 pixels as new_max_A for TVFA, AA and PA, respectively (relative scaling). The initial color of each variable is (255, 255, 20), (255, 20, 255) and (20, 255, 255). A circle of a radius of 10 pixels is used as the data point shape (DPS). The MLMV-GKU shows density clusters of both high and low density areas of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye.

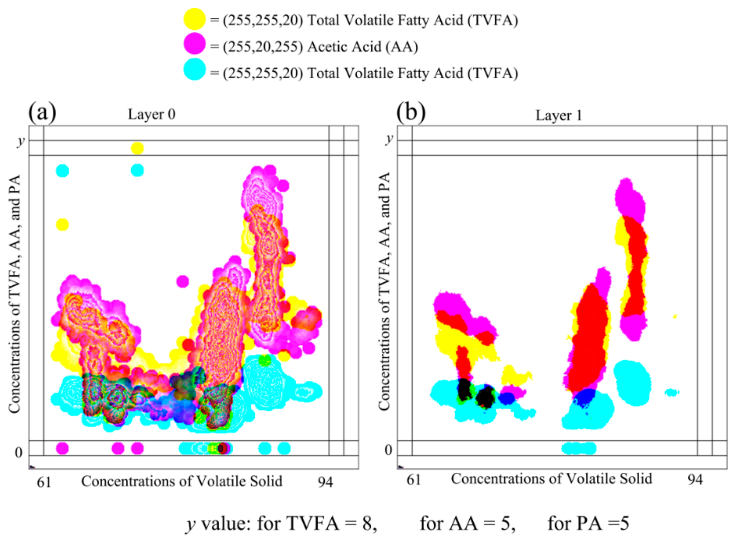

Figure 7. The MLMV-GKU of two layers. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. The effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method using 400 pixels as new_max_A for all dependent variables (absolute scaling). The initial color of each variable is (255, 255, 20), (255, 20, 255) and (20, 255, 255). A circle of a radius of 10 pixels is used as the data point shape (DPS). The MLMV-GKU shows the density clusters of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye.

Furthermore, Figure 6 shows a considerable amount of clusters between different variables. If there are clusters due to two or more variables, the color in the shared area is mixed and shows the clusters with a different color. With the basic knowledge of color formation theory, it is possible to identify those areas easily with the naked eye. Additionally, with image processing, it is also possible to identify these areas by analyzing the color values of pixels. When "Layer 1" is considered, there are no contour lines due to less overlapping. However, in Layer 1, clusters due to different variables can be easily identified due to the different color of the shared regions (e.g., the shared area of TVFA and AA). Furthermore, it is possible to distinguish clusters belonging to different variables. Since the place value of Layer 1 is higher than the place value of Layer 0, the presence of color in a higher layer implies the existence of higher density in the position in relation to the respective pixel. Therefore, the examination of higher layers makes higher density area identification much simpler.

Changing parameter values opens new ways of identifying new relations between different variables [27]. This technique is used as a very efficient tool in most data mining techniques. The MLMV-GKU also supports this feature and easily seeks relations by changing the overlapping between different variables. Figure 6 shows layers of the MLMV-GKU of the same dataset (scaled with the relative scaling technique), which is used in Figure 7, and scaled using "absolute scaling". The highest range among all of the dependent variables belongs to TVFA. Since R_{maxD} = 8, according to Equation (12), F_{iD} = 50 for TVFA, AA and PA. Then, according to Equation (13), new_max_A for TVFA, AA and PA is 400, 250 and 250, respectively.

Both layers in Figures 6 and 7 show overlapping areas of different variables. However, due to different scaling methods, the overlapping areas belong to different variables in Figures 6 and 7 which are not identical. This is a very useful feature when the GKU is used as a knowledge unit. When there is overlapping between variables, it is easy to identify color values of multiple parameters by analyzing a single pixel. This will reduce the overhead of computing. However, to interpret the meaning of a shared area due to one or more variables, sealing factors must be decided after properly investigating the domain requirements. Otherwise, only creating overlap areas between variables may not provide useful information.

Selecting different color schemes for representing the shape color of DPSs has considerable impact on the visual representation ability of GKU. In general, there are two available color schemes for representing three dependent variables using the 24-bit RGB pixel format as (255, 255, CI), (255, CI, 255), (CI, 255, 255) and (0, 0, CI), (0, CI, 0), (CI, 0, 0). Each color in the first version is near to the white color, which is represented as (255, 255, 255), and each color in the second version is near to the black color, which is represented as (0, 0, 0).

Figure 8 elaborates the impact of using the color scheme for the color of DPSs. Figure 8 contains the same data used in Figure 6 with the same scaling parameters. However, (0, 0, 20), (0, 20, 0) and (20, 0, 0) were used as colors of DPSs instead of (255, 255, 20), (255, 20, 255) and (20, 255, 255). The colors (255, 255, 20), (255, 20, 255) and (20, 255, 255) are not identical and are visually identifiable. Though colors (0, 0, 20), (0, 20, 0) and (20, 0, 0) are numerically identical, they are not visually identifiable because all of these colors are equivalent to the black color. The major drawback of this scheme is that the shape colors of DPSs cannot be visually identifiable, though they are different colors. This leads to difficulties in identifying clusters visually, especially in low density areas. When examining Layer 1 of Figure 8, this phenomenon is more visible. In Layer 1, neither clusters between different variables nor density clusters of individual variables are possible to identify. Furthermore, in Layer 0 of Figure 8, except high density areas, all of the other areas are in black or nearly black color, and this prevents visual identification of low density areas in relation to different variables. This phenomenon can be easily identified when considering the points P_A and P_B, as shown in Figure 8. It is not possible to distinguish between P_A and P_B visually due to the colors of the DPSs. However, Figure 6, these two points can be easily identified. Nevertheless, in reality, all black areas contain different colors (e.g., (0, 0, 40), (0, 40, 0) and (40, 0, 0)), which is visible only to an algorithm. Thus, selection of this type of color is not a problem for identifying clusters in low density areas with a computer. As a recommendation, it is stated that the different colors close to white are the best selection for the MLMV-GKU for visually capturing clusters between different variables, as well as density clusters of individual variables.

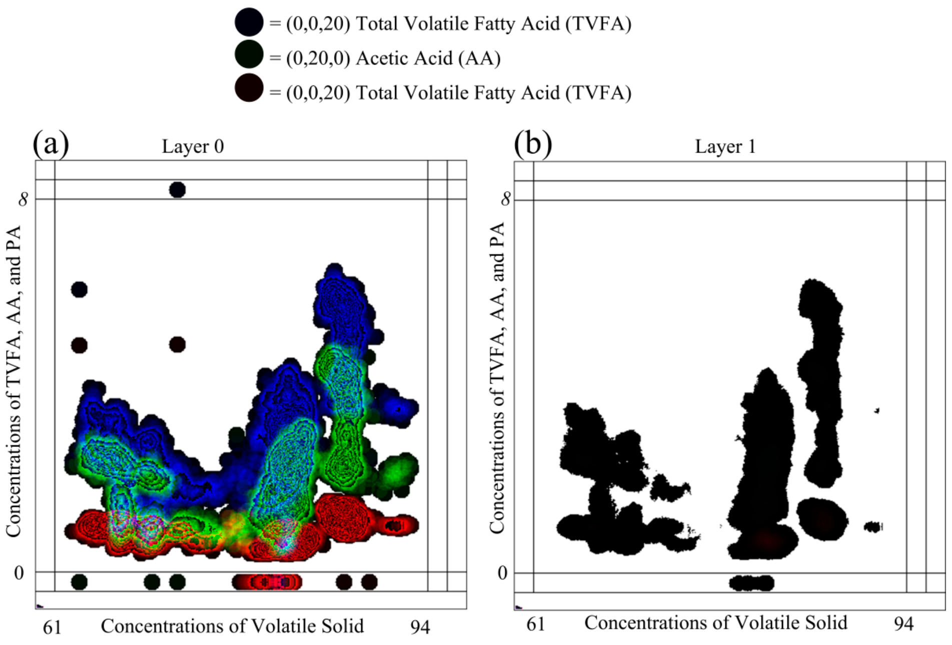

Figure 8. The MLMV-GKU representation of the same data shown in Figure 6 with a different color scheme. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. As in Figure 6, the effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method and used 400, 250 and 250 pixels as new_max_A for TVFA, AA and PA, respectively. The initial color of each variable is (0, 0, 20), (0, 20, 0) and (20, 0, 0). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows visual density clusters only in high overlapping areas. When there is low overlapping, it is not possible to distinguish points in relation to variables, such as P_A, P_B and Layer 0.

Plot (a) of Figure 9 shows the compression of the conventional scattered plot for one independent and three dependent variables and plot (b) of Figure 9 the MLMV-GKU of two layers for the same variables. The plots represent 21,951 data points, where the effective bitmap width and height are 400 and 625 pixels, respectively. Plots represent data in relation to three dependent variables (TVFA, AA and PA) against the concentration of volatile solids (VS). All of the data are normalized using the max-min normalization method and used 625, 550 and 225 pixels as new_max_A for TVFA, AA and PA, respectively. In the MLMV-GKU, the initial color of each variable is (255, 255, 10), (255, 10, 255) and (10, 255, 255). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows density clusters of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye. This is a special feature of the GKU, which is not possible with a scatter plot.

As mentioned in Section 1, star coordinates directly visualize dimensions as groups in a form of high density clusters in a two-dimensional plot. It is one of the better visualization techniques. However, star coordinates suffer from cluster overlapping of clusters of variables [17,20]. This is the major drawback of the star coordinates. If there are no overlapping regions, star coordinates can be used to visualize high density data clusters. However, the data we used have overlapping regions and cannot be visualized using star coordinates.

The nature of the MLMV-GKU allows the usage of a higher number of layers for representing variables. Unfortunately, this will lead to problems in understanding the whole picture. This is a very common problem in understanding scatterplot matrices [12]. To prevent this problem, it is recommended to use the maximum of four layers where the visual representation is important. Using four layers, it is possible to show 255 × 255 × 255 × 255 (more than four billion) overlaps using eight bits per variable, in the 32-bit pixel format. When using the 32-bit pixel format, basically, it is possible to show four dependent parameters against one independent parameter. This will provide space for more than 16 billion cases of overlapping (four billion × four variables) in a certain pixel coordinate of the four-layer GKU. If the width and height of the data plotting area are w pixels and h pixels, respectively, then it is possible to visualize the maximum of $x \times y \times 16$ billion cases of overlapping in the four-layer GKU. This is a huge amount of overlapping occurrences, and none of the existing graphical visualization techniques are capable of visualizing this amount of overlapping. Although it is recommended to use a maximum of four layers, there is no mathematical restriction to using more than four layers, depending on the requirement.

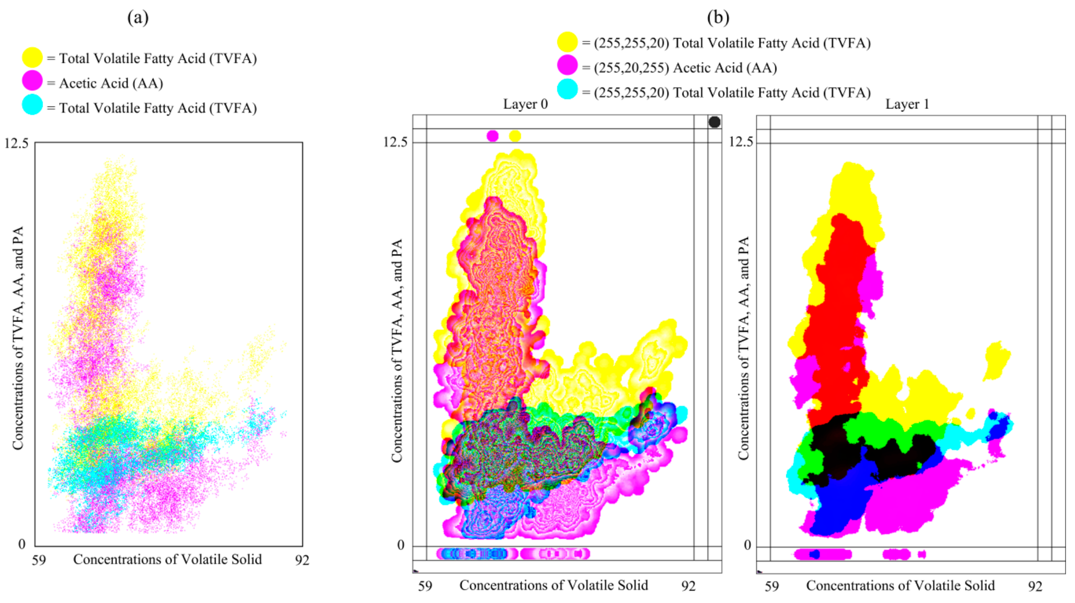

Figure 9. Scatter plot (**a**) and MLMV-GKU of two layers (**b**) representation of 21951 data points where effective bitmap width and height are 400 and 625 pixels. Plots representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids. All of the data points are normalized using the max-min normalization method and used 625, 550 and 225 pixels as new_max_A for TVFA, AA and PA, respectively. In the MLMV-GKU, the initial color of each variable is (255, 255, 10), (255, 10, 255) and (10, 255, 255). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows density clusters of both high and low density areas of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye, which is not possible with scatter plots.

The major drawback of MLMV-GKU is the influence of outliers, which leads to artificial high density areas on the GKU. Once these outliers are mixed with non-outlier values of other variables, this leads to incorrect decision making. Furthermore, due to the influence of outliers, the actual scaling cannot be achieved. P_A in Figure 8 can be considered as an example for such an outlier. Because of the influence of this outlier, it was considered as the maximum of the dataset, even though it is an outlier. Therefore, before plotting the MLMV-GKU, it is essential to remove outliers. The other disadvantage of the MLMV-GKU is that it cannot be directly used to find the correlation between variables, though sometimes, it helps to identify some trends between variables. Nevertheless, domain-related correlation can be defined after considering individual situations. For example, in the considered domain, the concentration of TVFA is proportional to the concentration of AA [29]. In Figure 6, clusters in relation to TVFA and AA have nearly the same shape and reveal the domain conditions more convincingly.

Dividing the 32 bits into small portions will facilitate using more variables in the MLMV-GKU. For example, if four bits are used per variable, it will provide room for eight dependent parameters to be clustered in the 32-bit format. Then, it is necessary to increase the number of layers to accommodate the higher number of data points. As previously mentioned, this is not always a good solution, especially if the aim is to use the MLMV-GKU for visualization. Therefore, the best solution is to use pixel formats that provide a higher number of bit ranges. There are different ARGB color formats that provide up to 128 bits per color channel (total 512 bits) [30,31]. Nevertheless, still, the limit of eight bits per variable and four layers for the MVML-GKU can be maintained.

The GKU is itself a database and knowledge unit that can be used as the input values for an algorithm; thus all of these properties are inherited by the MLMV-GKU, as well. The meaning of the content can be altered by changing three parameters of the DPS: firstly, the type of shape; secondly, the color of the shape; and thirdly, the position of the data point in the shape. The type of shape that is used in all of the plots in this article is circular. Furthermore, it is possible to use a suitable shape (e.g., square, polygon) instead of a circle. Furthermore, using different types of DPSs for different variables, it is possible to identify/show the effect/influence of a variable on other variables in a more meaningful manner. When a circle is selected with a single color as the DPS, this implies that the effect of the data point is equally valid for all of the pixels inside the circle. However, if the color of the circle is selected in a way that the color is proportional to a certain property (e.g., distance to the data point), this can be used to represent the functional influence of a data point. In the examples shown in this article, the data points were located in the center of the circle. By applying those property changes, the meaning of the MLMV-GKU can be enhanced according to different domain requirements.

If there is a dataset scheduled to be collected for a time period of T, at a time t_k $(t_k < T)$, all of the data placed on the MLMV-GKU are processed up to time t_{k-1}. Therefore, at any time, the data in the MLMV-GKU can be used to understand the current situation with or without further processing. If further processing is required, a copy of current bitmaps can be used for processing, while the original MLMV-GKU keeps updating. Because each layer is an independent bitmap, each layer can be individually analyzed using a separate process without depending on other layers. This feature enables the MLMV-GKU to be a parallel-processing-ready technique. After analyzing, the final decision can be obtained by summarizing the results of each layer. This could reduce the total processing time to below the usual single process analysis time.

7. Conclusions and Outlook

The MLMV-GKU is a technique that facilitates the visualization of multiple variables with a big amount of data. It does not show each data point individually and shows abstracted information as density clusters. Furthermore, the MLMV-GKU is a highly enhanced version of the GKU, which inherited all of the features of the GKU. The MLMV-GKU is an ideal tool for visualizing big data and fulfils a highly demanding requirement in the field of big data visualization. Using a four-layer GKU, it is possible to show more than four billion instances of overlapping using eight bits per variable, in the 32-bit pixel format. This will provide space for more than 16 billion different overlapping situations in a certain pixel coordinate. Thus, the MLMV-GKU will provide a fast means of decision making, which will be one of the major factors in process control. None of the existing graphical visualization techniques is capable of visualizing this amount of overlapping.

As an outlook for further developing the MLMV-GKU for the online environment, the usage of the 3D bitmap will facilitate the usage of another additional variable, the z-axis, in contrast to the current concept of the 2D bitmap, which facilitates $(1 + n)$ variables.

Acknowledgments: We are grateful to the German Academic Exchange Service (Deutscher Akademischer Austauschdienst) for providing a scholarship to K.K.L.B. Adikaram during the research period.

Author Contributions: Conceived of and designed the experiments and algorithms: K.K.L.B.A. Performed the experiments: K.K.L.B.A. Analyzed the data: K.K.L.B.A., M.A.H. and M.E. Contributed reagents/materials/analysis tools: K.K.L.B.A., M.A.H., M.E. and T.B. Wrote the paper: K.K.L.B.A.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Keim, D.A. Information visualization and visual data mining. *IEEE Trans. Vis. Computer Graph.* **2002**, *8*, 1–8. [CrossRef]
2. Cvek, U.; Trutschl, M.; Stone, R.; Syed, Z.; Clifford, J.L.; Sabichi, A.L. Multidimensional visualization tools for analysis of expression data. *World Acad. Sci. Eng. Technol.* **2009**, *30*, 281–289.
3. Barbará, D.; Chen, P. Using self-similarity to cluster large data sets. *Data Min. Knowl. Discov.* **2003**, *7*, 123–152. [CrossRef]
4. David, G.; Averbuch, A. Hierarchical data organization, clustering and denoising via localized diffusion folders. *Appl. Comput. Harmon Analy.* **2012**, *33*, 1–23. [CrossRef]
5. Galluccio, L.; Michel, O.; Comon, P.; Hero, A.O., III. Graph based K-means clustering. *Signal Process.* **2012**, *92*, 1970–1984. [CrossRef]
6. Chen, F.; Deng, P.; Wan, J.; Zhang, D.; Vasilakos, A.V.; Rong, X. Data mining for the internet of things: Literature review and challenges. *Int. J. Distrib. Sens. Netw.* **2015**, *501*. [CrossRef]
7. Chen, M.; Ebert, D.; Hagen, H.; Laramee, R.S.; van Liere, R.; Ma, K.L.; Ribarsky, W.; Scheuermann, G.; Silver, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.* **2009**, *29*, 12–19. [CrossRef] [PubMed]
8. Akerkar, R. *Big Data Computing*; CRC Press: Boca Raton, FL, USA, 2013.
9. Tsai, C.W.; Lai, C.F.; Chao, H.C.; Vasilakos, A. Big data analytics: A survey. *J. Big Data* **2015**, *2*. [CrossRef]
10. Karimi, H.A. *Big Data: Techniques and Technologies in Geoinformatics*; CRC Press: Boca Raton, FL, USA, 2014.
11. Fong, S.; Wong, R.; Vasilakos, A. Accelerated PSO swarm search feature selection for data stream mining big data. *IEEE Trans. Serv. Comput.* **2015**, *9*, 33–45. [CrossRef]
12. Chambers, J.M.; Cleveland, W.S.; Kleiner, B.; Tukey, P.A. *Graphical Methods for Data Analysis*; Wadsworth: Belmont, CA, USA, 1983.
13. Inselberg, A.; Dimsdale, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In Proceedings of the 1st Conference on Visualization, San Francisco, CA, USA, 23–26 October 1990; IEEE Computer Society Press: San Francisco, CA, USA, 1990; pp. 361–378.
14. Andrews, D.F. Plots of high-dimensional data. In *Biometrics*; International Biometric Society: Washington, DC, USA, 1972; pp. 125–136.
15. Bartke, K. 2D, 3D and High-Dimensional Data and Information Visualization. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.3421&rep=rep1&type=pdf (accesed on 28 March 2016).
16. Hoffman, P.; Grinstein, G.; Marx, K.; Grosse, I.; Stanley, E. DNA visual and analytic data mining. In Proceedings of the IEEE Visualization, Phoenix, AZ, USA, 24 October 1997; pp. 437–441.
17. Van Long, T.; Linsen, L. Visualizing high density clusters in multidimensional data using optimized star coordinates. *Comput. Stat.* **2011**, *26*, 655–678. [CrossRef]
18. Hoffman, P.; Grinstein, G. Visualizations for High Dimensional Data Mining-Table Visualizations. Available online: http://web.simmons.edu/~benoit/infovis/MIV-datamining.pdf (accessed on 28 January 2014).
19. Hoffman, P.E.; Grinstein, G.G. A survey of visualizations for high-dimensional data mining. In *Information Visualization in Data Mining and Knowledge Discovery*; Usama, F., Georges, G.G., Andreas, W., Eds.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2002; pp. 47–82.
20. Danyu, L.; Sprague, A.P.; Gray, J.G. PolyCluster: An interactive visualization approach to construct classification rules. In Proceedings of the 2004 International Conference on Machine Learning and Applications, Louisville, KY, USA, 16–18 December 2004; pp. 280–287.
21. Baraldi, A.N.; Enders, C.K. An introduction to modern missing data analyses. *J. Sch. Psychol.* **2010**, *48*, 5–37. [CrossRef] [PubMed]
22. Schafer, J.; Graham, J. Missing data: Our view of the state of the art. *Psychol. Methods* **2002**, *7*, 147–177. [CrossRef] [PubMed]

23. Nurunnabi, A.A.M.; Imon, A.H.M.R.; Ali, A.B.M.S.; Nasser, M. Outlier Detection in Linear Regression. In *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*; IGI Global: Hershey, PA, USA, 2011; pp. 510–550.
24. Beckman, R.J.; Cook, R.D. Outlier . . . . s. *Technometrics* **1983**, *25*, 119–149.
25. Adikaram, K.K.L.B.; Hussein, M.A.; Effenberger, M.; Becker, T. Continuous learning graphical knowledge unit for cluster identification in high density data sets. *IEEE Trans. Vis. Comput. Graph.* under review. **2016**.
26. Han, J.; Kamber, M.; Pei, J. *Data Mining, Southeast Asia Edition: Concepts and Techniques*; Elsevier Science: Amsterdam, The Netherlands, 2006.
27. Shalabi, L.A.; Shaaban, Z.; Kasasbeh, B. Data mining: A preprocessing engine. *J. Comput. Sci.* **2006**, *2*, 735–739. [CrossRef]
28. Krapf, L.C.; Heuwinkel, H.; Schmidhalter, U.; Gronauer, A. The potential for online monitoring of short-term process dynamics in anaerobic digestion using near-infrared spectroscopy. *Biomass Bioenergy* **2013**, *48*, 224–230. [CrossRef]
29. Gronauer, A.; Krapf, L.; Heuwinkel, H.; Schmidhalter, U. Near infrared spectroscopy calibrations for the estimation of process parameters of anaerobic digestion of energy crops and livestock residues. *J. Near Infrared Spectrosc.* **2011**, *19*, 479–493. [CrossRef]
30. Nikiel, S. *Iterated Function Systems for Real-Time Image Synthesis*; Springer: Berlin, Germany; Heidelberg, Germany, 2007.
31. Gelphman, D.; Laden, B. *Programming with Quartz: 2D and PDF Graphics in Mac OS X*; Elsevier Science: Amsterdam, The Netherlands, 2010.