Approximate Web Database Search Based on Euclidean Distance Measurement

Sarawuth Sonnum, Somtida Thaithieng, Sittichai Ano, Krerkchai Kusolchu,
and Nittaya Kerdprasop, *Member*, *IAENG*

***Abstract*—Although most methods for data finding and object searching are effective, they have one major weak point. That is they tend to eliminate a large number of objects whose properties do not sufficiently match those identified by the methods. As a result, a large number of possible choices are unnecessary eliminated. To fill this gap, we develop a new method that can include as many possibly relevant objects as possible to facilitate approximate search through the Web database. This method, based on the Euclidean distance measurement, finds objects with similar or closely properties and then identifies objects that share closest properties. This paper demonstrates how the method works by programming with the Erlang language to implement a Web-based search on mobile-phone application using Euclidean distance computation.**

***Index Terms*—Similarity search, Euclidian distance measurement, Web database, Erlang programming**

I. Introduction

EVEN though current technologies for data finding such as database search are highly effective, search results are normally limited to exact matching of the search predicates. However, in many cases users want to find some materials but they know only a rough characteristic of the objects. In such situations, exact finding may not be appropriate. Instead, approximate search should be a more suitable method. Similarity search is therefore a principal technique applied to the problem of approximate search.

Several researchers in the database community have studied similarity criteria in the database search. Agrawal, Faloutsos, and Swami [1] proposed the search technique in the context of sequence database. Chan and Fu [2] applied wavelet method to search patterns in the time series data. Faloutsos, Ranganathan, and Manolopoulos [3] also proposed a subsequent matching technique in the context of time series database. Subsequent work of Kahveci and Singh [6] and Shahabi and colleagues [7] were along the same direction of the previous proposed work.

We, on the contrary, study the similarity search problem in a rather different context. Our focus is on the approximate search of objects from the Web database with the intention to facilitate Web application. Out similarity criteria is based on the fundamental concept of Euclidean distance. By viewing data as a set of points in a multidimensional space, close distance between points suggests similarity between objects. The longer the distance, the less similar two objects are. If there is no separation between the objects (distance = 0), this implies that they are very similar objects.

The research work of Jeong and colleagues [5], Zhan and teammates [8] and Hui et al [4], are also based on the concept of Euclidean distance. But their work applied the concept for the different purposes of classification, data reduction, and data mining, respectively. The main purpose of our research is to devise an efficient similarity search based on closest Euclidean distance to extract the most similar objects to the users' preferences in the Web-based application.

II. Euclidean Distance Based Data Selection

A. Euclidian distance

In mathematics, the Euclidean distance or Euclidean metric is the ordinary distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric [9].

The Euclidean distance between points p and q is the length of the line segment, $\overline{pq}$. In Cartesian coordinates, if $p = (p_1, p_2,..., p_n)$ and $q = (q_1, q_2,..., q_n)$ are two points in Euclidean n-space, then the distance from p to q is given by the following equation.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Manuscript received December 16, 2010; revised January 20, 2011. This work is supported by School of Computer Engineering, Suranaree University of Technology, and the National Research Council of Thailand (NRCT).

S. Sonnum is a master student with the School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (e-mail: M5340316@g.sut.ac.th).

S. Thaithieng is a master student with the School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (e-mail: M5340323@g.sut.ac.th).

S. Ano is a master student with the School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (e-mail: casino_pao@hotmail.com).

K. Kusolchu is an assistant researcher at Suranaree University of Technology, Nakhon Ratchasima, Thailand.

N. Kerdprasop is an associate professor with the School of Computer Engineering and a researcher of the Data Engineering and Knowledge Discovery (DEKD) research unit, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: nittaya@sut.ac.th).

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. *Euclidian distance with nominal data*

This research provides methods for selecting data using Euclidian distance. From the equation we found that the appropriate data type for the calculation would be a numeric data type. In reality the data that we are considering will have some attributes in forms of non-numeric data type. Therefore, we have create an algorithm to find the distance value of the of the nominal data type as shown in Fig.1.

```
x = attribute value of target data.
y = attribute value of data being considered.

IF x == y THEN
    distance = 0
ELSE IF x != y THEN
    distance = 1
ENDIF
```

Fig. 1. Method to consider distance in nominal attributes

From the method given in Fig.1, we can then compute distance of the nominal data type. Here we have cataloged the nominal data into two values 0 and 1. The data that have distance = 0 mean that there is no distance between the data or in other word the data are the same. If the data have the distance = 1, that means the data are not the same. This method is similar to string matching in which the results would only be the same or not the same (true or false).

C. *Euclidian distance and data selection*

By looking at data in the form of the properties of the objects, we will see that each attribute of the data could correspond to each of the dimension on graph. When we plot each data record into the graph by the value of each attribute, we will get the data point as shown in Fig.2. We will find that if the data points on the graph are near each other, those data points will also show small differences of attribute values. Therefore, if we want data that is similar to one another, we only need to find the data points that are near the data point that we wanted. This principle has the same concept as the Euclidian distance equation which is to determine the similarity of the data by the search of minimum differences of the distance. If the differences between the distances are small, then the data are quite similar. Therefore, we use the Euclidian distance equation to find the selected data that we wanted.

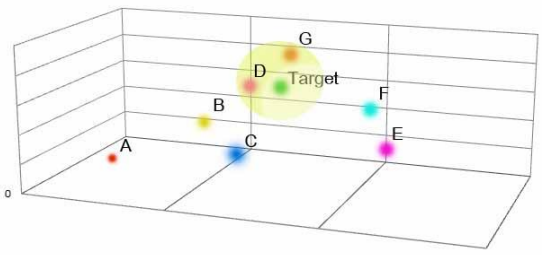

Fig. 2. Three dimensional graph that each dimension represents each attribute and each point represents each record of data

The example in Fig.2 shows a three dimensional graph, where each dimension represents each attribute of the data. The example contains a total of 7 records which are A, B, C, D, E, F and G. The location of the point where the data are represented is determined by the values of three variables. If the data values of the variable are similar, then the points will be neighboring to each other. In contrast, if the point locations are far apart, then the variables have great differences. When we plot the sample data (named target) into the chart, then we can find the information that is similar to the underlying data based on the distance between points. In this example, the D and G are the nearest points to the target.

III. Approximate Search Algorithm and Implementation

In this section, we will present an application of the Euclidian distance equation in finding similar data. The method in Fig.3 will be written in a form of natural language to make it easier to understand and be able to implement with any programming language. The steps in searching for similar objects are also provided in a flow chart form in Fig.4. We have presented an example on finding similar data of mobile phone where we will explant the process of finding the similar the data.

/* Euclidian based data selection algorithm */
1. Define value of each target attribute data.
2. Read comparative data form data source.
3. Calculate distance between target data and each record of comparative data.
4. Select the desired minimum threshold distance.
5. Select the data that has a value in the distance range

Fig. 3. The method of Euclidean based data selection algorithm.

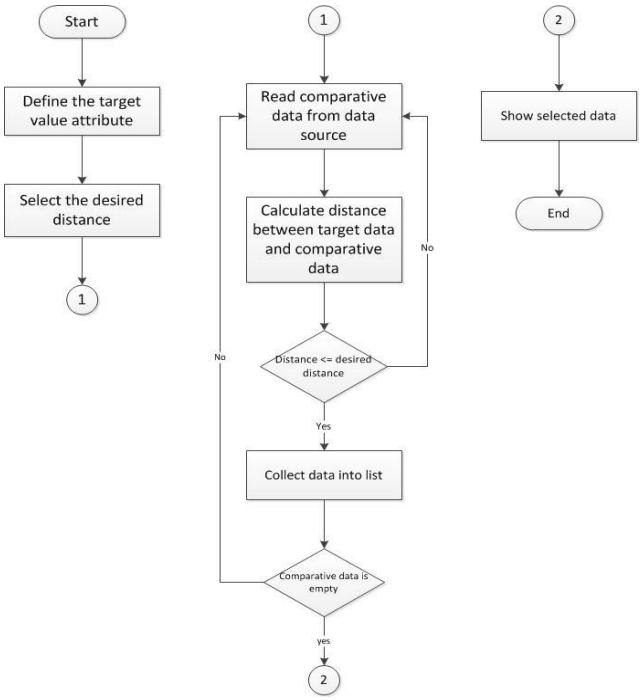

Fig. 4. Flow chart of Euclidian based data selection algorithm

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

To explain the idea of similarity search based on Euclidean distance, we use a Web database of mobile phone merchandises. The datasets are collected from the two available websites, which are www.thaimobilecenter.com and www.siamphone.com. The data collected from both websites are the detail of mobile phones sold through the internet. Each phone is described by 10 attributes which are the model, name, price, resolution, width, height, thickness, camera resolution, touch screen capability, and operating system. Each mobile phone is thus represented as a data point in a 10-dimensional space. Data are stored in a mobile data source (Fig. 5). Users can then search for desired mobile phone through the Web application. The process in approximate similarity search is explained via running examples in the following subsections.

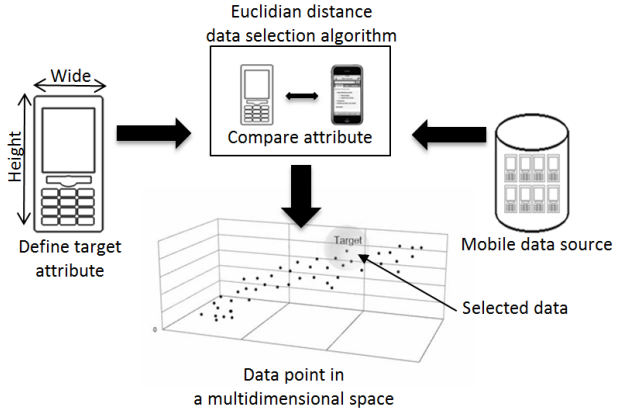

Fig. 5. Samples of Euclidian data selection algorithm for mobile selection that each data record will plot into multidimensional space

A. *Euclidean data selection example*

TABLE I
Sample data

Model	Resolution (pixel)	Width (mm.)	Height (mm.)	Weight (gram)
Mobile 1	65536	62	112.2	132
Mobile 2	144000	50	108	130
Mobile 3	65000	62	112	122
Mobile 4	408000	74	132	160
Mobile 5	65536	60	112	124
Mobile 6	172800	65	122.4	130
Mobile 7	65536	63	132	127
Mobile 8	144000	50	130	140

To illustrate our proposed method of approximate search for objects closest to the interest of users, we provide a simple database as shown in Table I. The sample database contains eight records of mobile phones with five attributes. The steps in finding specific mobile phones are as follows.

1) The target object p specified by users is a mobile phone with pixel resolution = 65000 pixels, Width = 60 mm, Height = 110 mm, Weight = 120 grams.

2) For each data record q in the database, find the mobile phones that are nearest to the target object using Euclidean distance measurement.

$$d(p,q) = \sqrt{\sum_{i=1}^{8}(p_i - q_i)^2}$$

The computed distances between the target object and the eight data records are displayed in Table II.

TABLE II
Distance form target data

Model	Distance
Mobile 1	**536.14**
Mobile 2	79000.00
Mobile 3	**3.46**
Mobile 4	343000.00
Mobile 5	**536.02**
Mobile 6	107800.00
Mobile 7	**536.50**
Mobile 8	79000.02

3) Select the nearest data record by expected distance. For example, suppose the user identifies that the distance must less than 1000. Then the selected data in this case are Mobile 1, 3, 5, and 7. These are the mobile phones closely correspond to the user's interest

In the example as shown above, it is the rough selection of the mobile phone preferences that we have did in advance. Now we want to find the distance between the target data and each data and the distance will be the indicator of the difference in the similarity. If the value of distance is a lot, that means that the object will be very different. On the other hand, if the value of distance is very small and closer to 0, this would mean that this record is very similar to the target data and if the value is 0, then that means the two data sets are the same or are very similar.

From the example, we can notice that to set the distance for each type of data it needs to be set appropriately to the data. We must consider the number taken into consideration. If the number is large, then the distance will be far away as well. Note that if we want to use this principle to other data, then the data must also have the appropriate distance. In addition, it is also found that selecting data by this method, we will need to compare data by computing the distance of every data record. If there is a lot of information in the database, then this method would become very slow. In such case, it may be necessary to reduce the data by configuring select variables as the main data attributes to be considered. We will thus rely on primary data in filtering out unnecessary information which would make the method run faster. This principle will be presented through the next example.

B. *Master key data method*

TABLE III
Sample data with master key

Model	Resolution (pixel)	Camera resolution (pixel)	OS
Mobile A	65536	3200000	Android
Mobile B	144000	5000000	Bada
Mobile C	172800	2000000	Meego
Mobile D	65000	3200000	Android
Mobile E	408000	3200000	Android
Mobile F	408000	5000000	Meego
Mobile G	65000	5000000	Limo
Mobile H	65536	2000000	BlackBerry

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1) Consider the sample database given in Table III. Select main value attribute (master key); in this example chose OS as main key and Android as main value. Then prepare data by delete some records that their values are not equal to the main value. The result is shown in Table IV

TABLE IV
Prepared data

Model	Resolution (pixel)	Camera resolution (pixel)	OS
Mobile A	65536	3200000	Android
Mobile D	65000	3200000	Android
Mobile E	408000	3200000	Android

2) Select data by Euclidean data selection algorithm.

In this example, the auxiliary step will be the data preparation for data selection, which can be done by general DBMS principle that is to select the specific record which we want to select and do not consider the other information. This selection step can be done with the following SQL command.

SELECT * FROM TABLE_IV WHERE OS = 'Android'

Selecting the information in this manner can be done very quickly. The result is a database that has some records cut out. When used in the process of selecting data using Euclidean data selection algorithm, this database will reduce processing time. In this example, we will find that the information that will be used as the master key should be a nominal data type. This is because the data is divided into clear group which is perfect to be used as a main key in selecting data. If we use a numeric data type as the master key, then the data may be selected by the range of data representation. The information obtained will be the reduced number of records to be processed. This will decrease the processing time.

C. Algorithm implementation

In this subsection, we demonstrate the implementation of Euclidean distance based data selection algorithm using Erlang as a programming language. Erlang is a functional language with a declarative style of function declaration. This allows program implementation to be done in a short time. Its data structures represented as tuples and lists are very efficient to manipulate data. Fig. 6 shows the information of telecommunication Web databases that was transformed to be in a list format of the Erlang language. The sample data are stored in a data.dat file.

```
[12500,65536,60,109,13.9,104,2000000,1,4].
[21900,172800,62,111,14.6,161.1,5000000,1,4].
[0,144000,50,108,13.3,93,3200000,1,4].
[13500,172800,62,112,14,136,3200000,1,4].
[21400,65536,62,112.2,13.9,160,3200000,1,4].
[22900,65536,59.96,108.45,14.27,120,3200000,1,4].
[12900,65000,62,112,14.2,130,3200000,1,4].
[25500,65000,60,109,14,110,3200000,1,4].
[13900,65000,60,108.8,13.9,106,2000000,1,4].
[29900,65000,62.2,112.5,13.95,155,3200000,1,4].
[11500,65536,50,101,17.5,102,2000000,1,4].
[25500,65000,65.8,113.8,65.8,133,2000000,1,4].
[19900,65000,50,107,14,91,2000000,1,4].
[25000,65000,60,107,15.5,111,2000000,1,4].
[27000,65000,66,114,14,134,0,1,4].
[26000,65000,66,114,14,134,0,1,4].
[18000,65000,50,107,14.5,89.5,0,1,4].
[27900,65000,69.5,110,19.5,134,0,1,4].
[24900,65000,74.5,113,22,139,0,1,4].
[8900,262144,59.5,116,12.44,117,5000000,1,3].
[0,408000,63,109.5,13.3,122,3000000,1,3].
[0,408000,60,119,11.7,134,5000000,1,3].
[0,408000,61,97,16.7,145,3000000,1,3].
[21000,172800,60,109,14,112,5000000,1,4].
[12500,65536,60,109,13.9,104,2000000,1,4].
[21900,172800,62,111,14.6,161.1,5000000,1,4].
[0,144000,50,108,13.3,93,3200000,1,4].
[13500,172800,62,112,14,136,3200000,1,4].
[21400,65536,62,112.2,13.9,160,3200000,1,4].
[22900,65536,59.96,108.45,14.27,120,3200000,1,4].
[12900,65000,62,112,14.2,130,3200000,1,4].
[25500,65000,60,109,14,110,3200000,1,4].
[13900,65000,60,108.8,13.9,106,2000000,1,4].
[29900,65000,62.2,112.5,13.95,155,3200000,1,4].
[11500,65536,50,101,17.5,102,2000000,1,4].
[25500,65000,65.8,113.8,65.8,133,2000000,1,4].
```

Fig. 6. Sample of data list that prepared to process with Erlang

The program coding of our proposed algorithm is provided in Fig. 7. The function find is the main function of our approximate search program. The user has to specify target object characteristics as the parameters of the function find. The program also asks for the desired search distance. The running example of our Erlang program is given in Fig.8.

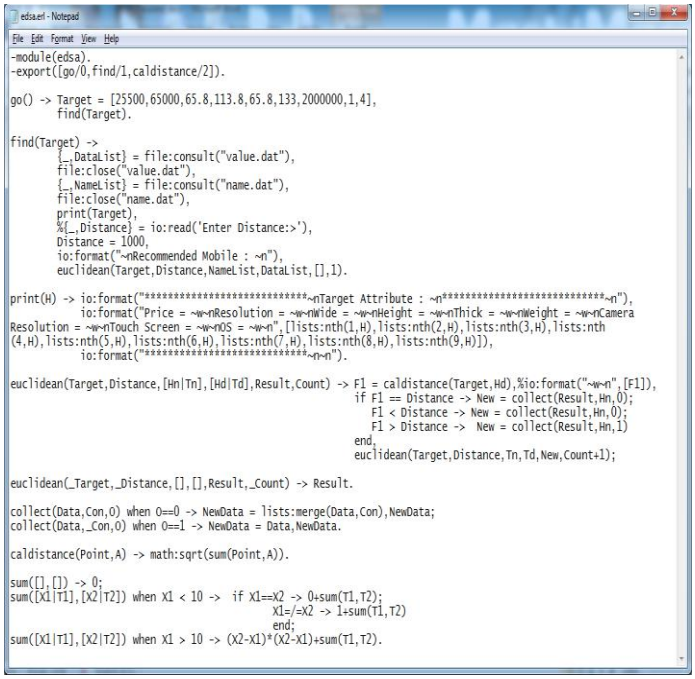

Fig. 7 Program coding with the Erlang language

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

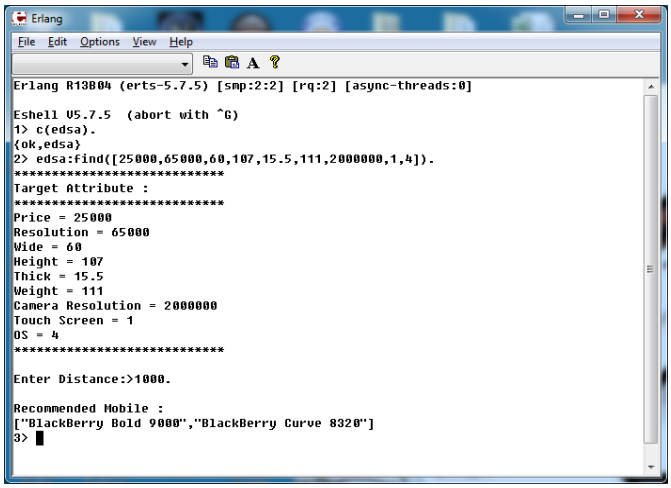

Fig. 8. Screenshot of running prepared data with Erlang

IV. Experimentation

In the running example as shown in Fig. 8, we search the mobile product with a set of parameters, that is, price, resolution, width, height, thickness, weight, camera resolution, touch screen capability, and operating system. By specifying the desired distance as 1000, the search results are two mobile phones: Blackberry Bold 9000 and Blackberry Curve 8320. These approximate search results are correct.

The next step of our experimentation is to test the program performance on large data sizes. We prepare 7 databases with different number of data records. The test databases contain 300, 600, 1200, 2400, 4800, 9600, and 19200 records. The real data collected from the websites (www.thaimobilecenter.com and www.siamphone.com) consist of 2700 records. We thus duplicate the number of data records to the specified amount. The running time of our Erlang program on each database is graphically shown in Fig. 9.

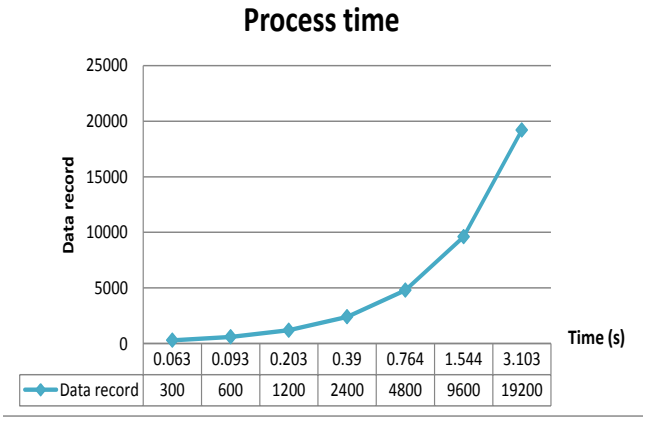

Fig. 9. Processing time of Erlang with different amount of data

From the processing time results, it is found that when the amount of data increases, the running time of the search program also increases correspondingly. The increase in processing time is exponential. However, searching for target object in the large database with 19200 records takes computational time of only 3 seconds. This running time is acceptable. But for a larger database with data more than hundreds of thousand records, a more efficient search method like parallel computation is necessary.

V. Conclusion

This paper presents the approximate search method based on the Euclidian distance concept. The purpose of our research is to find the most similar objects to the users' preferences in the Web-based applications. The users' queries identify some attributes of the desired objects. Our algorithm then computes the Euclidian distances of the target object and the surrounding data. From the given distance threshold, the algorithm can produce the most relevant objects to the user interest.

We implement the proposed method with the Erlang programming language and test the program with the telecommunication Web database. The experimental results reveal that the program can display similar objects within the short period of time. The proposed method can thus be applied to other Web applications.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," In FODO, Evanston, Illinois, October 1993.
[2] K.-P. Chan and A.W.-C. Fu. Efficient time series matching by wavelets. In ICDE, 1999.
[3] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series database," In SIGMOD, pages 419–429, Minneapolis MN, May 1994.
[4] Y. Hui, W. Yufang and G. Yuan, "Research on similarity mining for flight data," Second International Workshop on Education Technology and Computer Science, 2010.
[5] S. Jeong,S.-W.Kim and B.-U.Choi, "An optimization algorithm of K-NH classification," Springer Science + Business Media, LLC 2008,October 2008.
[6] T. Kahveci and A. Singh, "Variable length queries for time series data," In ICDE, Heidelberg, Germany, 2001.
[7] C. Shahabi, X. Tian, and W. Zhao, "TSA-tree: A wavelet-based approach to improve the efficiency of multi-level surprise and trend queries," In SSDBM, 2000.
[8] Y. Zhan, H. Chen and G. Zhang, "Dimensionality reduction for similarity search with the Euclidean distance in high-dimensional applications," Fifth International Conference on Machine Learning and Cybernetics, Dalian, August 2006.
[9] Euclidean distance measurement, access on 5 December 2010, http://en.wikipedia.org/wiki/Euclidean_ distance

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)