Efficient Retrieval on Dense Vector by Similarity Preserve Hash in Vegetable Geographical Origin Identification System

Nobuyoshi Sato[†], *Minoru Uehara*[††], *Koichiro Shimomura*[†], *Hirobumi Yamamoto*[†] *and Kenichi Kamijo*[†]

[†]Plant Regulation Research Center, Toyo University, Itakura-machi, Gunma 374-0193 Japan
[††]Department of Information and Computer Sciences, Toyo University, Kawagoe City, Saitama 350-8585 Japan

Summary
Recently, camouflaging geographical origin of agricultural products is a major problem in Japan. Therefore, we developed a distributed geographical origin identification system which identifies cultivated farms of vegetables by using trace element compositions of vegetables. This system stores compositions of trace, or very small quantities of elements into database which located on farming districts, and compares them to trace element compositions of vegetables which gathered from food distribution channel such as markets, food factories. The comparison is done by calculating correlation coefficient. This system can be considered as an information retrieval system which gives only an answer of which trace element composition is similar to given query vegetable's trace element composition. In this system, trace element compositions are expressed as dense vector, and they are retrieved by one-to-one comparison. In this paper, we describe applying Similarity Preserve Hash (SPH) into our geographical origin identification system, with assumption on data disposition.

Key words:
Similarity Preserve Hash (SPH), Retrieval on dense vector.

1. Introduction

Recently, food safety problems such as BSE, camouflaging cultivated place of vegetables is a big issue in Japan. To cope with this problem, some food traceability systems are now partly on practical use in Japan. In food traceability systems, IDs such as barcodes or RFID tags into packages. However, there is an important problem that traceability systems chase only IDs. Therefore, ID/package switching/forging cases have been exposed in Japan.

In such situations, we proposed a distributed system which identifies geographical origin of vegetables by analyzing, accumulating and comparing trace element compositions of vegetables[1][2]. This system is able to identify cultivated farms of vegetables by vegetables themselves.

Plants such as vegetables absorb metal ions in the soil. Since compositions of metal ions in the soil differs from geographical places, even farms, and fields. Therefore, compositions of metal ions in the soil and vegetables which absorbed them have different features, it is thought. In our system, databases are located in farms, agricultural organizations and so on in every farming district. Trace element compositions of shipped vegetables are recorded onto these databases, and they are compared to trace element compositions of vegetables gathered from distribution channel such as food factories, wholesale and retail markets. Unlike food traceability system, our system analyzes and compares vegetables themselves. So, deceiving our system is difficult, we thought.

Our geographical origin identification system can be considered as a sort of information retrieval system that gives only an answer for a given query. Trace element compositions are expressed as high dimensional dense vector in our system. Generally, efficient method on all of storing data, retrieving data, and storage space usage without any assumptions character of data itself. As a very initial implementation, our system employed one-to-one comparison by calculating correlation coefficients between given query vegetable's trace element vector and all stored trace element vector. So response time to identify was slow. Therefore, in this paper, we introduce to employ Similarity Preserve Hash (SPH)[3] to improve response time to identify geographical origin of a vegetable. SPH is a simple idea based hash function that outputs similar values for similar inputs.

The organization of this paper is as follows. In section 2, as related works, we will describe some works on food safety problem and SPH. In section 3, we describe outline of our system. In section 4, we will describe a method to accelerate response time to identify geographical origin by reduction of the number of one-to-one comparison to find similar trace element composition data. Finally, we will conclude.

Manuscript received October 11, 2006.
Manuscript revised October 25, 2006.

2. Related Works

2.1 Chemists' works

Many chemists have challenged to identify geographical origin in wine[4], coffee[5], tea[6], potatoes [7]and orange juice[8], especially, geographical origin identification of wine are researched in Europe from almost 20 years ago. In these researches, not only trace elements analysis but also abundance of stable isotopes and differences on ingredient of organic matter are used to identify. On the other hand, in Japan, geographical identification by trace elements analysis in Japanese leeks (welsh onions)[9], onions[10], garlic[11] and unpolished rice[12] are already researched. However, although researches on identification by trace elements analysis showed possibility of discrimination, but geographical identification is not yet. Therefore, to realize geographical identification, accumulating trace elements analysis data of each producing districts are needed. However, geographical and temporal granularity of accumulating data is not clear now, so we cannot decide that what kinds of data should be stored. Also, relations between parts of a vegetable, circumstance of soil and fertilization and trace elements is not clear now. However, in general terms, there are some cases such as Japanese leeks cultivated in Japan and China seem to be distinguishable, a cultivated products are grown in Japan or other continent is distinguishable by using abundance of stable isotopes which is related with ages of soil is created. Thus, geographical origin identification by trace elements is promising way, we think.

2.2 Similarity Preserve Hash

Since trace element compositions are expressed dense vector with 6 or more dimensions, efficient indexing method is needed to accelerate response time of the system when identifying. R-tree and its derivations and so on are famous as methods to index high dimensional data. In high dimensional data, there is no known method which satisfies all of storage space efficiency, cost for creating index, cost to retrieval, and no assumptions on data. However, this also means that there are some efficient methods except one of storage space efficiency, indexing cost, retrieval cost and assumptions on data. Generally, R-tree and derivations are not friendly to existing relational database and applications. Although PostgreSQL can make R-tree index, however, this is supported on only few geometric data type such as box, circle. Therefore, we employed a method which is not high dimensional tree based.

Similarity Preserve Hash (SPH)[3] is a hash function based on vector space model which gives similar output for similar input, unlike ordinary hash functions. Normally, ordinary hash functions do not depend on length of input, however, input for SPH should be any fixed length to best use of feature of SPH. That is, the input for SPH should be m dimensions vector. In SPH, to get n bits of output, n random vectors are prepared. If random vectors were changed, the output of SPH will change. Therefore, changing random vector after once data accumulation is started will have expensive cost.

To establish standard of SPH output, n random vectors $\boldsymbol{R}_n = (r_{n,1}\ r_{n,2}\ \ldots\ r_{n,m})$ in m dimension space are prepared. Let a vector for input of SPH be $\boldsymbol{S}=(s_1\ s_2\ \ldots\ s_m)$. An angle between $\boldsymbol{R}_n$ and $\boldsymbol{S}$ is calculated as follows:

$$d_n = \frac{\sum (s_m - \bar{s})(r_{n,m} - \bar{r}_n)}{\sqrt{\sum (s_m - \bar{s})^2}\sqrt{\sum (r_{n,m} - \bar{r}_n)^2}} \quad (1)$$

Here, $\bar{s}$ is average of $(s_1\ s_2\ \ldots\ s_m)$, $\bar{r}_n$ is average of $(r_{n,1}\ r_{n,2}\ \ldots\ r_{n,m})$. h_n which is the n-th bit of SPH hash value for input $\boldsymbol{S}$ is defined using d_n as follows:

$$h_n = \begin{cases} 1 & d_n \geq 0 \\ 0 & d_n < 0 \end{cases} \quad (2)$$

And, these series of bits are assembled into SPH hash value as $h_{n-1}h_{n-2}\ldots h_2h_1h_0$.

In our geographical origin identification system, all trace element data to calculate correlation coefficients are standardized. So $\boldsymbol{R}_n$ are generated so that probability distribution of each element of $\boldsymbol{R}_\mathbf{n}$ is Gaussian distribution.

To employ SPH to retrieve a vector, a SPH value of a query vector $\boldsymbol{Q}$, SPH($\boldsymbol{Q}$)is used. However, we must care that SPH values of vectors $\boldsymbol{R}$ SPH($\boldsymbol{R}$) which should hit to $\boldsymbol{Q}$ do not have always the same value to SPH($\boldsymbol{Q}$), except for in case of $\boldsymbol{Q}$ and $\boldsymbol{R}$ are exactly the same. That is, all SPH($\boldsymbol{R}$)s which are within few bit Hamming distance from SPH($\boldsymbol{Q}$) must be searched. When a SPH hash bit string has k bits width and l bits are selected for Hamming, the number of all SPH($\boldsymbol{R}$) bit strings a which are within l bit Hamming distance is $a = \sum_{i=1}^{l}{}_k C_i$. The number of combination explodes on value of k and l. We confirmed that this can be reduced to $a = \sum_{i=1}^{l}{}_{\frac{k}{2}} C_i$. Detail will be described in Section 4.

3. Geographical Origin Identification System

At first, we describe our proposed method to identify geographical origins of vegetables using their trace element compositions. In our method, trace element composition of both shipped vegetables from and gathered vegetables on food distribution channel are compared as vector forms, and if angle of two vector is almost 0, corresponding farm is judged as a geographical origin.

Trace element compositions are standardized. And, angle of vectors are calculated as correlation coefficient.

A vector of m sorts elements $\boldsymbol{S}_i = (s_{i,1}\ s_{i,2}\ \ldots\ s_{i,m})$ express trace element compositions of vegetables which shipped and measured. Let i be an ID of shipped vegetable. Here, let trace element compositions of vegetables gathered from food distribution channel be $\boldsymbol{U} = (u_1\ u_2\ \ldots\ u_m)$, correlation coefficient of two vectors $r_{\boldsymbol{S}_i,\boldsymbol{U}}$ is as follows:

$$r_{S_i,U} = \frac{\sum (s_{i,m} - \bar{s}_i)(u_m - \bar{u})}{\sqrt{\sum (s_{i,m} - \bar{s}_i)^2}\sqrt{\sum (u_m - \bar{u})^2}} \quad (3)$$

When $r_{\boldsymbol{S}_i,\boldsymbol{U}}$ overcomes a suitable threshold near 1, $\boldsymbol{S}_i$ and $\boldsymbol{U}$ are considered that they are grown in the same field of a farm. Note that there may exists some $\boldsymbol{S}_i$ that overcomes the threshold, and not always $\boldsymbol{S}_i$ which have the maximum $r_{\boldsymbol{S}_i,\boldsymbol{U}}$ is the geographical origin.

To prevent possibility of iniquity, it is desirable that trace element compositions of vegetables are measured, stored at agricultural organizations on faming districts. So we decided to locate distributed databases which accumulate trace element compositions onto farming districts. Fig.1 shows the overview of proposing system. Databases which accumulate trace element compositions are located onto agricultural organizations in each district as DataSite. Furthermore, to cope to iniquity by agricultural organization on each district, a ControleSite is installed in an agricultural authority organization.

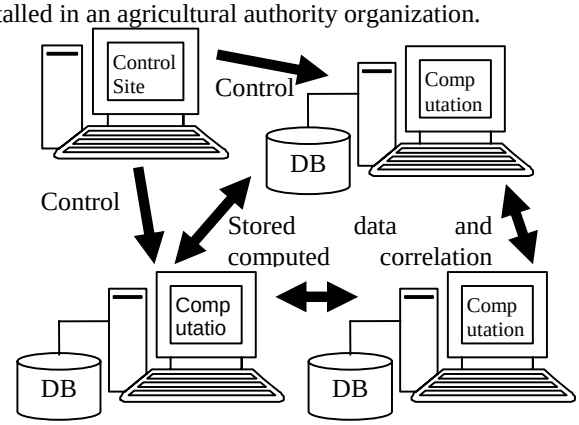

Fig.1 The overview of geographical origin identification system of vegetables

4. Query Target Selection by SPH

As described above, our proposing system identifies geographical origin of vegetables by calculating correlation coefficients. Generally, since a correlation coefficient of two vectors cannot be known before it is calculated, our system employed one-to-one comparison between a query vector and all trace element composition vectors of vegetables stored in databases in Data Site. That is, calculating correlation coefficients itself is one-to-one comparison. Since there are currently about two million farms in Japan[13], we estimate that hundred thousands of farms cultivate the same crop and breeds. Since we have an idea to install geographical origin identification kiosk for consumers into all supermarkets, response time of identification system should be as short as possible to realize interactive kiosk.

Here, we define response time of our system to identify geographical origin as sum of three steps; data retrieval time from database, calculation time of correlation coefficients, and transfer time of calculation results between Data Sites and Control Site. Table 1 shows that the response time of our geographical origin identification system when a query is given by a user in our previous work[2]. All of retrieval time, calculation time and transfer time seem to be $O(n)$ against the number of stored samples. All programs for evaluation in this paper are written in Perl, except a program for computation is written in C. FreeBSD 5.5R, PostgreSQL 7.4.2, Pentium4 2.8GHz PC is used for all evaluations. All retrieval target data is simulated data based on actual trace element composition measured on our previous works[1][2].

Firstly, we tried to reduce the number of one-to-one comparison to reduce computation time and CPU load to calculate correlation coefficients. This can be realized by rough selection of targets to calculate correlation coefficients by SPH, as described in section 4.1. Next, we tried to reduce retrieval time that increased by employing SPH. Transfer time can be reduced by limiting correlation coefficients of calculation results to be transferred. However, this will be realized automatically by previous two efforts.

Table 1 Response time in case of one-to-one comparison

#stored samples	Retrieval time [sec.]	Computation time [sec.]	Transfer time [m:s]
1000	0.27	0.56	0:01.49
10000	1.37	1.82	0:08.96
100000	12.13	15.13	1:05.67

4.1 Fundamentals of SPH and Naive Method

Since output of SPH depends on random vectors, random vectors must be prepared before accumulating data onto database and they should not be changed once a data is stored.

Firstly, a set of random vector $\boldsymbol{R}$ is generated by ControlSite, and distributed to all DataSite. In our system, correlation coefficients are calculated in standardized vector space. Therefore, $\boldsymbol{R}$ is randomly generated so that

they accords Gaussian distribution. This means that random vectors have no need to be adjusted to averages and standard deviations of trace element composition data.

Trace element data is standardized by following steps; calculation of averages of each element through all farms, calculating standard deviations of them, and standardization itself.

Averages of each element

Average of composition of an element in site *i*, x_{i*e} is as follows:

$$\bar{x}_{i*e} = \frac{1}{m_i}\sum_{j=1}^{m_i} x_{ije} \tag{4}$$

Here, x_{ije} represents composition of element *e* of stored sample in *j*-th on site *i*. m_i represents the number of stored items in site *i*. Each site can calculate $\bar{x}_{ije}$ without communicating to other site. Average of *e* in entire the site is as follows:

$$\bar{x}_{**e} = \frac{1}{\sum_{i=1}^{n} m_i}\sum_{i=1}^{n} m_i \bar{x}_{i*e} \tag{5}$$

Here, *n* is the number of sites.

Standard deviations of each element

Standard deviation s_e of entire the sites can be calculated as follows:

$$\begin{aligned} s_e &= \sqrt{\frac{1}{\sum_{i=1}^{n} m_i - 1}\sum_{i=1}^{n}\sum_{j=1}^{m_i}(x_{ije} - \bar{x}_{**e})^2} \\ &= \sqrt{\frac{1}{\sum_{i=1}^{n} m_i - 1}\sum_{i=1}^{n} d_{ie}} \end{aligned} \tag{6}$$

Here, $d_{ie} = \sum_{j=1}^{m_i}(x_{ije} - \bar{x}_{**e})^2$ is residual sum of squares of *e* in site *i*. Each site calculates d_{ie}, ControlSite gathers d_{ie}, and calculates s_e.

Standardization and calculation of SPH

Next, stored data are standardized and their SPH are calculated as following:

1. Let standardized vector of *j*-th data in site *i* be $\boldsymbol{Y}_{ij}=(y_{ij1}\ y_{ij2}\ \ldots\ y_{ijl})$, *e* be element. Each element of standardized vector y_{ije} can be calculated with:
$$y_{ije} = \frac{x_{ije} - \bar{x}_{**e}}{s_e} \tag{7}$$
$\boldsymbol{Y}_{ij}$ can be calculated in each DataSite. Here, *l* is the number of element of stored data, or dimension of vectors.
2. DataSite *i* calculates SPH hash value of $\boldsymbol{Y}_{ij}$ based on $\boldsymbol{R}$ $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Y}_{ij})$. In case of new $\boldsymbol{R}$ is generated by ControlSite, DataSite *i* calculates $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Y}_{ij})$ of all *j*. When *j* is newly added, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Y}_{ij})$ for newly added *j* is calculated.
3. DataSite *i* sends $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Y}_{ij})$ to ControlSite. Useful attributes of stored data for target selection such as shipped date is sent at the same time.

Since averages and standard deviations are calculated from stored data, x_{**e} and s_e can vary by adding data to database. Therefore, x_{**e} and s_e should be always recalculated when a trace element data is added. Since recalculation of x_{**e} and s_e brings recalculation of SPH, this is very ineffective. However, if the number of stored trace element data is enough large and all farming districts are already registered before calculating x_{**e} and s_e, it is thought that x_{**e} and s_e do not vary so much. So frequent recalculation of x_{**e} and s_e can be avoided. We think it is enough that x_{**e} and s_e are recalculated a time in a year for each harvesting season.

Geographical origin identification procedure

Next, we describe behavior of the system to identify geographical origin of a vegetable gathered from food distribution channel.

1. A client for identification *c* calculated SPH hash value of a identifying sample $\boldsymbol{U}=(u_1\ u_2\ \ldots\ u_l)$, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U})$. Here, $\boldsymbol{U}$ is standardized in advance.
2. c sends $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U})$ with crop, breed and shipping date of identifying sample to ControlSite, and request to target selection.
3. ControlSite enumerates a set *S* of DataSite *i* having data which consists $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U})$ and $\mathrm{Hamming}_z(\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U}))$, and sends *S* to *c*. Here, $\mathrm{Hamming}_z(\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U}))$ is a set of all SPH which are within *z* bit of Hamming distance from $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U})$.
4. *c* obtains standardized trace element compositions Y_{ij} which satisfy $\mathrm{Hamming}_z(\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{U}))$.
5. *c* calculates correlation coefficients between identifying sample and obtained data from each site as follows:
$$r_{(Y_{ij}U)} = \frac{\sum_{e=1}^{l}(u_e - \bar{u})(y_{ije} - \bar{y}_{ij*})}{\sqrt{\sum_{e=1}^{l}(u_e - \bar{u})^2}\sqrt{\sum_{e=1}^{l}(y_{ije} - \bar{y}_{ij*})^2}} \tag{8}$$
6. *c* treats combination of *i* and *j* those $\boldsymbol{r}_{(\boldsymbol{Y}_{ij}\boldsymbol{U})}$ overcomes a threshold *t* as result of identification. If there are some combinations of *i* and *j*, *c* treats combinations of *i* and *j* those shipping date, breed and geographical origin is the same to identifying sample as proper. Otherwise, all combinations are shown to user.

Here, we describe a method to reduce the number of Hamming SPH strings by restricting bit selected for Hamming. Fig.2 shows a simplified example to calculate 8 bit SPH. In Fig.2, $\boldsymbol{Q}$ is a vector of identifying sample's query vector, $\boldsymbol{A}$ and $\boldsymbol{B}$ are vector of trace element

compositions of stored samples, $\boldsymbol{R}_0$ though $\boldsymbol{R}_7$ are random vectors to calculate SPH.

As described before, *n*-th bit of SPH of a vector $\boldsymbol{Q}$ is calculated following when correlation coefficients between $\boldsymbol{Q}$ and $\boldsymbol{R}_0$ through $\boldsymbol{R}_7$ are expressed as d_n:

$$h_n = \begin{cases} 1 & d_n \geq 0 \\ 0 & d_n < 0 \end{cases} \quad (9)$$

That is, h_n is 1 when an angle between $\boldsymbol{Q}$ and $\boldsymbol{R}_n$ is lesser than or equal to 90°, h_n is 0 otherwise. A SPH is assembled as a series of bits $h_7h_6h_5h_4h_3h_2h_1h_0$. Therefore, in example of Fig.2, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Q})$ is 01110101. On a vector $\boldsymbol{A}$ near $\boldsymbol{Q}$, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{A})$ is 01110101, exactly matches to $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Q})$. However, on another vector $\boldsymbol{B}$ near $\boldsymbol{Q}$, of which distance to $\boldsymbol{Q}$ is similar to distance from $\boldsymbol{A}$ to $\boldsymbol{Q}$, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{B})$ is 01110001, not matches $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{Q})$. On $\boldsymbol{B}$, a bit h_2 calculated from $\boldsymbol{R}_2$ differs, however, $\boldsymbol{B}$ has almost the same importance to $\boldsymbol{A}$ as vectors near $\boldsymbol{Q}$, and $\boldsymbol{B}$ must not be leaked from retrieval results for geographical origin identification.

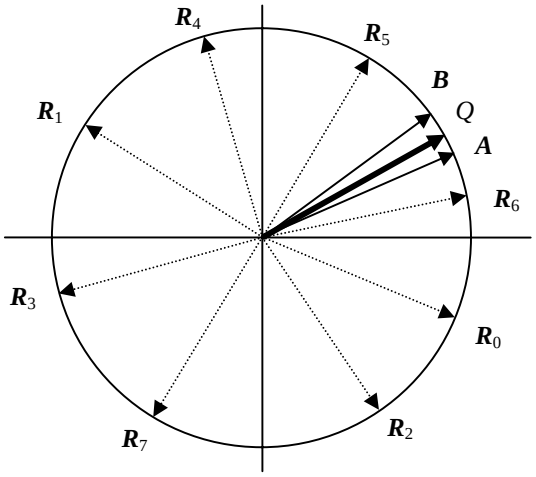

Fig.2 A simplified view of SPH

Here, $\mathrm{SPH}_{\boldsymbol{R}}(\boldsymbol{B})$ differs because d_2 is negative for $\boldsymbol{B}$. This brings an idea to give higher priority to $\boldsymbol{R}_n$ those sign is easily reversed, in different words, $\boldsymbol{R}_n$ which crosses almost right angles to $\boldsymbol{Q}$. On the other hand, vectors do not cross right angles (in Fig.2, $\boldsymbol{R}_3$ and $\boldsymbol{R}_6$ etc.) have no meanings to be selected for Hamming bit. Therefore, it has high efficiency when Hamming bits are selected in increasing order of d_n. The number of combination of all Hamming bit strings for a query a will be $a = \sum_{x=1}^{h} {}_n C_x + 1$ when width of SPH is n (or n random vector is prepared), and h bits are chosen from it. And, a can be reduced. To select Hamming bits with this way, firstly Hamming candidate bits are selected from entire bits of a SPH, then, few bits for Hamming are selected from these candidate bits.

4.2 Evaluation of Naive Method

Fig. 3 explains that relation ship between the number of Hamming bits to be searched and the number of hits for each query. Here, we retrieved 100,000 queries from 100,000 stored trace element data. Both of queries and stored data have 6 elements, and are randomly generated based on Gaussian distribution. A set of 32 random vectors with 6 dimensions were used. In Fig.3, "3 bit Hamming" means that all combination of SPH Hamming bit strings within 3 bit Hamming distance from query's SPH of each query were searched. Threshold of correlation coefficients which judges whether particular samples match to given query was 0.990. In Fig.3, a curve titled "All targets are searched" explains that the number of hit items in case of 100% recall ratio. This means that it is ideal case if a SPH employed result have the same or much closed curve to "All targets are searched". At the result, "4 bits Hamming" have very closed curve to it, the runner-up is "3 bits Hamming". So it seem to be enough in case of all SPH bit strings within 3 or 4 bits Hamming distance from given query's SPH are searched.

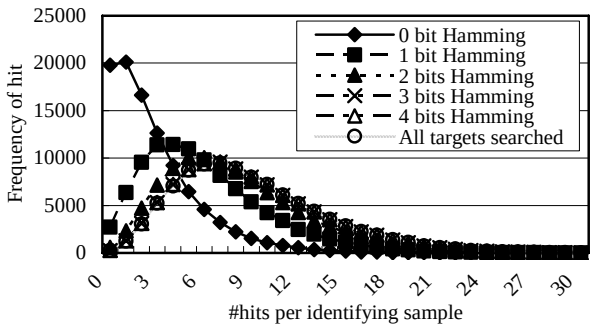

Fig. 3 Histogram of #hits for each #Hamming bits
100000 queries are retrieved against 100000 stored data

Fig. 4 shows that frequency of the number of hits for each Hamming bit string. Here, all of 9,277,285 bit strings within 3 bit Hamming distance from given 100,000 queries are searched. Over 8,829,027 Hamming SPH bit strings have no hit. This means that there are only few Hamming SPH bit strings which hit to given query, within *n* bits Hamming distance from SPH of given query. So this wastefulness must be removed for practical use.

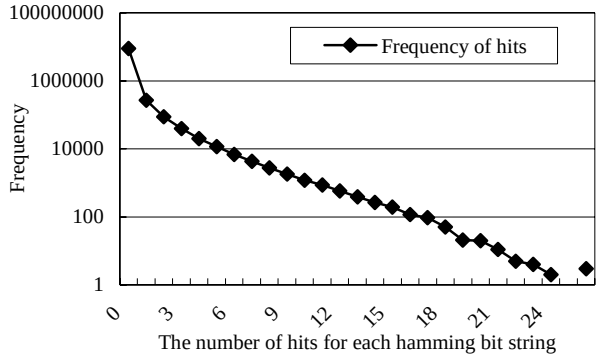

Fig.4 #hits per Hamming bits and frequency 100000 queries are retrieved against 100000 items of stored data

Here, we describe evaluations about a method to make retrieval efficiency by restricting Hamming bits of query SPH and reducing the number of Hamming SPH bit strings. Fig.5 shows that the number of hits per each Hamming SPH. Here, each bit of query's SPH is sorted in increase order of absolute values of correlation coefficient d_n between query $\boldsymbol{Q}$ and random vectors $\boldsymbol{R}_n$. In this experiment, 100,000 queries are used for 100,000 stored trace element composition data, as same as above experiment. The number of SPH bits is also 32. The number of hits decreases from order 1 to 5 linearly, and on order 10 and upper, only few hits are counted. The lowest order which hit was 15 when 3 bits are used for Hamming, 13 for 3 bits Hamming, and its number of hit was 1. As this result, in retrieval using SPH, it is enough that only 13 or 15 bits which have low absolute value of d_n are selected for Hamming. In this case, only 16 bits those $\boldsymbol{R}_n$ is close to $\boldsymbol{Q}$ should be selected for margin is considered. Also, in cases of the number of bits of SPH is increased, it is thought that almost half bits of SPH is selected for Hamming, this will be enough if random vectors are generated completely random. This means that the number of all combination of Hamming bit strings can be reduced to $\sum_{x=1}^{h} {}_{n/2}C_x + 1$.

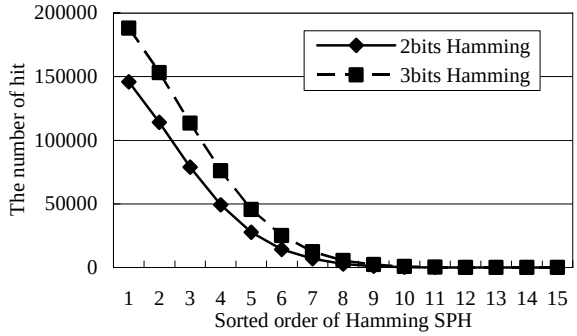

Fig.5 #hits of Hamming SPH sorted in order of distance between Hamming SPH bits and query's SPH

Table 2 shows response time when SPH is employed, both in case of Hamming bits are restricted to 16 bits, and not restricted. Computation time and transfer time are reduced drastically. They seem to be $O(1)$ against the number of stored trace element data in database in both two cases. However, retrieval time become so long And retrieval time seem to be $O(n)$ against the number of stored trace element data. Fig.6 shows detailed retrieval time. Unfortunately, retrieval time seem to be $O(n^2)$ against the number of Hamming bit. Since all SPH bit strings within 3 (or even 4 in unfortunate case) bits Hamming distance should be searched to avoid leakage of retrieval in Fig.3, this slow retrieval time is not acceptable. However, in overall saying, SPH successfully reduced computation time and transfer time. Therefore, we tried to reduce retrieval time as described following subsection.

Table 2 Response time when 3 bit Hamming SPH is employed

(a) Hamming bit restriction is not employed

#stored samples	Retrieval time [sec.]	Computation time [sec.]	Transfer time[sec.]
1000	2.53	0.00013	0.63
10000	22.83	0.00016	0.64
100000	237.39	0.00057	0.71

(b) Hamming bit restriction is employed

#stored samples	Retrieval time [sec.]	Computation time [sec.]	Transfer time[sec.]
1000	0.27	0.00012	0.64
10000	2.17	0.00013	0.63
100000	19.75	0.00019	0.65

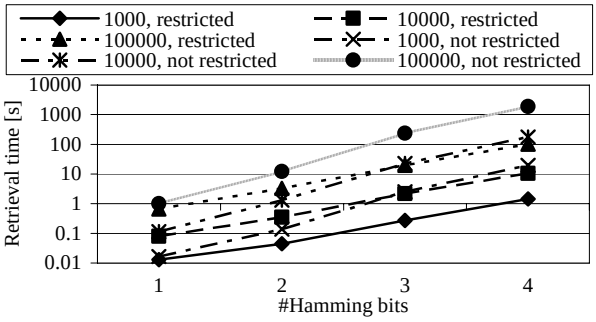

Fig.6 #Hamming bits, #stored trace element data vs. retrieval time

4.3 Acceleration retrieval time of SPH by grouping

Here, firstly, we discuss about why retrieval time increased when SPH is employed. Data is obtained from database by using all Hamming bit strings of SPH as key. Therefore, in the SQL query, all Hamming bit strings are

enumerated like SELECT * FROM trace_elem_table WHERE sph IN (0x01234567, 0x01234568, … 0x13568013);. If value of Hamming bit strings continues, BETWEEN can be used to grouping of these bit strings. So grouping by BETWEEN is able to used when Hamming bits are chosen from LSB in sequence. However, since Hamming bits are chosen by order of angle between given query and random vectors. Therefore, if Hamming bits satisfied this condition (to a certain extent) on Hamming order, it was just a coincidence. When all humming bit strings which have Hamming distance within 3 bits form given query's 32 bits SPH are enumerated, the number items on a set of Hamming bit sequence is $\sum_{x=1}^{h} {}_{n/2}C_x + 1 = 697$, when h=3 as described before. Table 3 shows execution time of SELECT statement when all bit sequences are enumerated in WHERE condition.

Table 3 Execution and invocation time vs. the number of bit sequences when they are enumerated in WHERE on SELECT. statement

#Hamming bits	Exec. time [m:s]		#bit sequences	
	32 bit	16 bit	32 bit	16 bit
1	0:01.2	0.00.7	33	17
2	0:13.3	0:05.4	529	137
3	4:39.7	0:34.1	5489	697
4	32:41.7	1:56.0	41449	2517

Response time when this SELECT statement is executed is thought as $O(n)$ where n means the number of bit sequences enumerated in WHERE condition. Therefore, it is important that to reduce the number of members on WHERE condition to shorten response time.

To reduce the number of members on WHERE condition, grouping of SPH is one of possible way. Since SPH puts similar output for similar input, this way is thought as promising. In following, we will introduce and compare some grouping methods.

There three methods for grouping of SPH.

- Use value of several bits of MSB of SPH. In this paper, we examined when MSB 4 bits are used. Thus, all values of SPH and Hamming bit sequences are divided into 16 groups. In WHERE condition, bit sequences of MSB are enumerated.
- Transform SPH onto Gray code, and its several bits of MSB is used for grouping. Here, MSB 4 bits are used the same as above.
- Grouping by the number of set bits on a bit sequence of a particular SPH. The number of set bits on Hamming bit sequence is always the number of set bits on original SPH±n bits. In this paper, to divide into 16 groups, we employed one half of the number of set bits.

The group information is stored before selecting target for calculation of correlation coefficients. That is, a column which expresses MSB and the number of set bits of SPH is added onto the table of trace element data. When retrieving targets, the column is used on WHERE condition. The number of groups enumerated on WHERE condition is 16 in the maximum. This is extremely smaller than when Hamming bit sequences of SPH are enumerated directly. When retrieving, data items of which SPH exactly matches to query's SPH and its Hamming bit sequences must be extracted from the result of SELECT statement. In this way, rapid reduction of working set can be realized.

4.4 Evaluation of Grouping SPH Method

Table 4 shows that execution time of a query of SELECT statement and extract data of which SPH exactly matches. In this table, execution time includes invocation of a program, inquire to database and extraction. Here, the number of Hamming bits for SPH is with in 3 bits. These three results are 10 times faster than a result on the same Hamming condition of 0:34.1[m:s] in Table 3, when Hamming target bits are restricted to 16 bits and 3 bits of them are used to Hamming. There is no difference observed in three methods, however the number of bits method takes little longer. Causes of shorten of execution time can be thought as following: shorten of disk access time by reduction of the number of members enumerated in WHERE condition, execution to pick up data of which SPH exactly matches to given query's SPH and its Hamming bit sequence set is done completely on memory.

Table 4 Execution time of SELECT statement when 3 grouping methods are used, Hamming bits are restricted to 16 bits, 3 bits are selected for Hamming

Grouping method	Exec. time of SELECT
MSB	3.31 [sec.]
Gray code	3.14 [sec.]
The number of set bits	3.64 [sec.]

Table 5 shows that execution time when 1000 queries are retrieved by these three methods. Only the number of bits method takes over 40 minutes, and other ends in 30 minutes.

Table 5 Execution time when 1000 queries are retrieved, in the same condition to Table 4

	Exec. time of SELECT
MSB	25:09 [m:s]
Gray code	27:24 [m:s]
The number of set bits	41:44 [m:s]

Fig. 7 shows that frequency of the number of items divided in a particular group on three grouping methods. Here, the number of Hamming bits is 3, and the frequency of each group is sorted as decreasing order. The number of set bits method has always only 4 groups. Other methods have up to 8 groups.

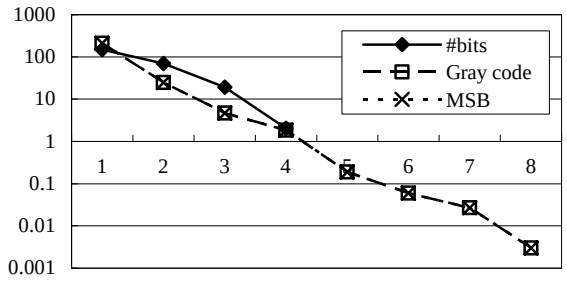

Fig.7 Frequency of the number of items divided in groups by three grouping methods

Next, we describe efficiency of target selection by these grouping methods. Objective of grouping is to realize rapid reduction of the size of working set, by rough selection of data of which SPH exactly matches to query's SPH and its Hamming bit sequences. Therefore, efficiency of the grouping methods must be evaluated by size of reduced working set, and ratio of aim data on the reduced working set. Table 6 shows that sizes of reduced working sets on three methods. Here, 1000 queries are retrieved against trace element database which has 100000 data. The number of data which exactly matches query's SPH and its Hamming bit strings was 204.2 for each query in average. In methods on MSB and Gray codes, working the size of working set is reduced to 1/3.

As the bottom line, MSB or Gray code method is efficient. To consider facility of implement, MSB could be employed in our distributed geographical origin identification system for green groceries.

Table 6 Sizes of reduced working set on three methods

Grouping methods	Size of working set [items]
MSB	32005
Gray code	32005
The number set bits	61003

Finally, Table 7 shows that improved response time of our geographical origin identification system when SPH grouping MSB method is employed, compared to Table 1. Retrieval time seem to become constant against the number of stored samples. Although computation time and transfer time is $O(n)$ against the number of items selected by SPH grouping method in principal, computation time and transfer time seem to be almost constant against the number of stored samples.

Table 7 Response times when acceleration by SPH grouping MSB method is employed

#stored samples	Retrieval time [sec.]	Computation time [sec.]	Transfer time[sec.]
1000	3.13	0.0042	0.67
10000	3.63	0.0043	0.64
100000	3.31	0.0060	0.65

5. Conclusions

In this paper, we described how to employ SPH in our geographical origin identification system. Unlike R-tree and other high dimensional index methods, SPH is friendly to generally used relational database systems. Our initial implementation employed one-to-one comparison by calculating correlation coefficient to identify cultivated farms of vegetables. Therefore, we employed SPH in our system, and reduced the number of one-to-one comparison by calculating correlation coefficients. This realized significant reduction of computation time. However, since all bit series within 3 or 4 bits must be retrieved from database, and the bit series are enumerated in SELECT statement, retrieval time become slower. So, next, we improved retrieval time by grouping SPH. This made retrieval time as acceptable short time. This work made the response time about 4 seconds. Although the response time when the number of stored samples is few became slower, however, response time when large numbers of data are stored in database became significantly faster, and the response time seem to be almost $O(1)$ against the number of stored samples.

Acknowledgments

This work was supported by "University-Industry Joint Research" Project for Private Universities; subsidy by MEXT (Ministry of Education, Culture, Sports, Science and Technology), 2003-2007.

References

[1] Nobuyoshi Sato, Minoru Uehara, Jin Tamaoka, Koichiro Shimomura, Hirobumi Yamamoto, Kenichi Kamijo, "Initial Design of Distributed Identification System for Geographical Origin by Trace Element Analysis," in *Proc. of 8th International Workshop on Network-based Information Systems*, pp.79-83. (2005)

[2] Nobuyoshi Sato, Minoru Uehara, Jin Tamaoka, Koichiro Shimomura, Hirobumi Yamamoto, Kenichi Kamijo, "A Distributed Geographical Origin Identification System for Agricultural Products by Trace Element Compositions", *International Journal of Computer Science and Network Security*, Vol.5, No.10, pp.55-63. (2005)

[3] Moses S. Charikar, "Similarity Estimation Techniques from Rounding Algorithms", in *Proc. of 4th Annual ACM Symposium on Theory of Computing*, pp.380-388. (2002)

[4] Malcom J. Baxter, Helen M. Crews, M. John Dennis, Ian Goodall, Dorothy Anderson, "The determination of the authenticity of wine from its trace element composition", *Food Chemistry*, Vol.60, No.3, pp.443-450 (1997)

[5] Kim A. Anderson, Brian W. Smith, "Chemical Profiling to Differentiate Geographic Growing Origins of Coffee", *J. Agric. Food Chem.*, Vol.50, No.7, pp.2068-2075 (2002)
[6] Pedro L. Fernádez-Cáceres, María J. Martín, Fernando Pablos, A. Gustavo González, "Differentiation of Tea (Camellia sinensis) Varieties and Their Geographical Origin According to their Metal Content, *J. Agric. Food Chem.*, Vol.49, No.10, pp.4775-4779 (2001)
[7] Kim A. Anderson, Bernadene A. Magnuson, Matther L. Tschirgi, Brian Smith, "Determining the Geographical Origin of Potatoes with Trace Metal Analysis Using Statistical and Neural Network Classifiers", *J. Agric. Food Chem.*, Vol.47, No.4, pp.1568-1575 (1999)
[8] Wayne A. Simpkins, Honway Louie, Michael Wu, Mark Harrison, David Goldberg, "Trace elements in Australian orange juice and other products", *Food Chemistry*, Vol.71, No.4, pp.423-433 (2000)
[9] Kaoru Ariyama, Hiroshi Horita, Akemi Yasui, "Chemometric Techiques on Inorganic Elements Composition for the Determination of the Geographical Origin of Welsh Onions", *Analytical Sciences*, Vol.20, No.5, pp.871-877 (2004)
[10] Yoji Taguchi, "Discrimination of green-groceries by analyzing composition and quantity of inorganic elements —Inorganic elements analysis by ICP-MS and an estimate of cultivated places of onions—", I*nvestigation Research Report of IAA Center for Food Quality, Labeling and Consumer Services*, No.26, http://www.cfqlcs.go.jp/technical_information/investigation_research_report/pdf/2601.pdf (2002) (in Japanese)
[11] Masahiro Yasui, Keiko Kinoshita, Daio Kozuka, Yusuke Kitani, Yoshinori Yasui, Ken Takubo, Harue Saito, Masaaki Morita, "Analysis method to mortgage reasonable indication of cultivated place —Comparison of garlic cultivated in Japan and China— ", *Investigation Research Report of IAA Center for Food Quality, Labeling and Consumer Services*, No.22, http://www.cfqlcs.go.jp/technical_information/investigation_research_report/pdf/2209.pdf (1998) (in Japanese)
[12] Akemi Yasui, Kumiko Shinodh, "Determination of the geographic origin of brown-rice with trace-element composition", *Bunseki Kakagu*, Vol.49, No.6, pp.406-410. (2000) (in Japanese)
[13] Statistics Bureau & Statistical Research and Training Institute, Ministry of Internal Affairs and Communications, "*Historical Statistics of Japan*", http://www.stat.go.jp/english/data/chouki/

Nobuyoshi Sato was born in 1976. He received his Ph.D. degree in Computer Science from Toyo University, Japan in 2004. He is currently a research assistant at Plant Regulation Research Center, Toyo University. His research interests include distributed system, information retrieval and WWW applications. He is a member of IEEE, IEICE and IPSJ.

Minoru Uehara was born in 1964. He received his Ph.D. degree in Computer Science from Keio University in 1995. He is currently a professor at Department of Information and Computer Sciences, Toyo University. His research interests include distributed system, and programming language. He is a member of IEEE, ACM, and IPSJ.

Koichiro Shimomura was born in 1951. He received his Ph.D. degree in Pharmacy in 1981 from Kyushu University. He joined National Institute of Health Sciences. He is now a professor at Faculty of Life Sciences, Toyo University from 2000. His research interest is mainly antioxidative compounds produced by plants. He is a member of Pharmaceutical Society of Japan, Japan Society for Bioscience, Biotechnology, and Agrochemistry and Japanese Society for Plant Cell and Molecular Biology.

Hirobumi Yamamoto was born in 1960. He received his Ph.D. degree in Pharmacy in 1989 from Kyoto University. He was an assistant professor in Faculty of Pharmaceutical Sciences, Nagasaki University. He is now a professor at Faculty of Life Sciences, Toyo University from 2003. His research interests are biochemistry and metabolic engineering in plant. He is a member of Pharmaceutical Society of Japan and Japanese Society for Plant Cell and Molecular Biology.

Kenichi Kamijo was born in 1949. He received his Ph.D. degree in Geophysics from Kyoto University in 1994. He is now professor at Faculty of Life Sciences, Toyo University. His research interests include complex systems in informatics, geoinformatics and bioinformatics. He is a member of IEICE, JSAI, Meteorological Society of Japan and Geodetic Society of Japan.