修　士　論　文

Judgment assistance of search engine results by using the Batch-Learning Self-Organizing Map algorithm

指導教官　Ho Tu Bao　教授

北陸先端科学技術大学院大学
知識科学研究科知識システム学専攻

150049　中本　修

審査委員：Ho Tu Bao 教授（主査）
石崎 雅人 助教授
佐藤 賢二 助教授
林 幸雄　 助教授

２００３年２月

Contents

Figure contents

Table contents

Chapter1. Introduction

1.1. Research background

The development of World Wide Web(WWW) offers us various information. In recent years, in the information society it becomes the necessity and indispensable. However, information open to the public in WWW is huge. Therefore, it is difficult to retrieve information that we need efficiently.

The search engine of the robot type, for example, Google[4] and the directory type search engine , for example, yahoo[5] are used to collect information which is necessary usually. The Web document, which corresponds to necessary information by the user's inputting the key word, and filtering the Web document is obtained.

Recently, the robot type search engine is chiefly used. Because the volume and update speed of data of the robot type search engine is more excellent than that of the directory type search engine. But it can be said that it is difficult to look for information is necessary from the result of the robot type search engine. The reason is that the retrieval result is displayed without being made to the classification.

There is two timing that the retrieval result classifies. One is a method of the classification before, and another one is a method of the classification after. The classification before is done by the directory type search engine.

It thinks the classification after it retrieves it. There is a classification according to an absolute standard. There is a classification relatively done. A relative classification will be done in this thesis. There is Self-Organization Map(SOM) as a means to classify, and to output from the feature of input information. However, SOM has the feature depends in the order of inputting data. I want to treat any data equally this time. Then, Batch-Learning Self-Organization Map (BLSOM) which removes the input order dependence from SOM is applied.

Namazu was used for the robot type search engine in the thesis. The reason for this is the full-text search engine that Namazu is the most famous. And

1.2. Thesis composition

This thesis becomes the following compositions. Chapter 3 explains and discusses mounting this system and the performance evaluation about the research which relates to a basic matter in Chapter 2. Chapter 4 describes the conclusion and the problem of this thesis

Chapter2.Fundamental matter and relation research

2.1.Fundamental matter

2.1.1.Search engine

2.1.1.1.Kind of search engine

There are two kinds of engines of the robot type and the directory type in the inside though it is known as a search engine usually

The robot type search engine is a type which automatically goes round Web by the called program such as crawler/Spider (crawler at the following), collects the web pages, and makes the data base.

The representative of this type is “Google” “goo[6]” etc.

The merit of the robot type search engine is that the update speed with a lot of volume of information is fast. The reason for this is that the data collection is done by the automatic operation.

Because all pages on the web are retrieved, information with low quality might be included in the retrieval result oppositely as a weak point. However, it is almost lost that the page with low quality is displayed in the high rank recently by the accuracy improvement of the retrieval algorithm.

Only the site admitted that "Directory type search engine" those who inspect it about man who is called editor/surfer etc. examine the website, and is profitable for the user is registered in the data base.

The representative of this type is "Yahoo Japan. " “BT Looksmart[7]" etc.

The merit of the directory type search engine is that there is a lot of information with a generally high-quality registered site. The reason for this is that the examination with person's hand is done to the registration of the site.

The weak point is that volume of information is a little oppositely.

The purpose of this is for registration to rely on all person's hands. Also in the robot type search engine and the directory type search engine, there are strengths and limitations respectively, and it is managed in shape to supplement them in many cases each other.

For instance, Yahoo At retrievals by keyword Yahoo in case of Japan When target information is not found in data base of Japan, the key word is succeeded to Google. And the retrieval result from the data base of Google and is Yahoo It is displayed as page retrieval result of Japan.

Moreover, Google uses the data of Open Directory Project as a directory. nfoseek has the directory of independence in shape to supplement the robot type retrieval.

2.1.1.2. Management process of robot type search engine

Fig 1is the one that the management process of the robot search engine was made figure. The crawler automatically goes round the WEB document, and the indexer and the called program make data base/index (index at the following) for the robot type search engine.

A lot of robot type search engines request the word at this time and the cutting out key word appearance frequency etc. are requested. Additionally, each word analyzes whether the mark improvement is done with what kind of tag, it arranges to easy-to-use shape for each search engine, and it stores it in the index.

And, it retrieves in the data base according to each retrieval algorithm when retrieval key/Ceri (retrieval key at the following) is given, and URL is listed in order with high agreement level to the given retrieval key.

Thus, there are two or more stages in the management process of the robot type search engine, and it begins to differ to the difference of the retrieval result of each search engine about the technology which takes charge of of each.

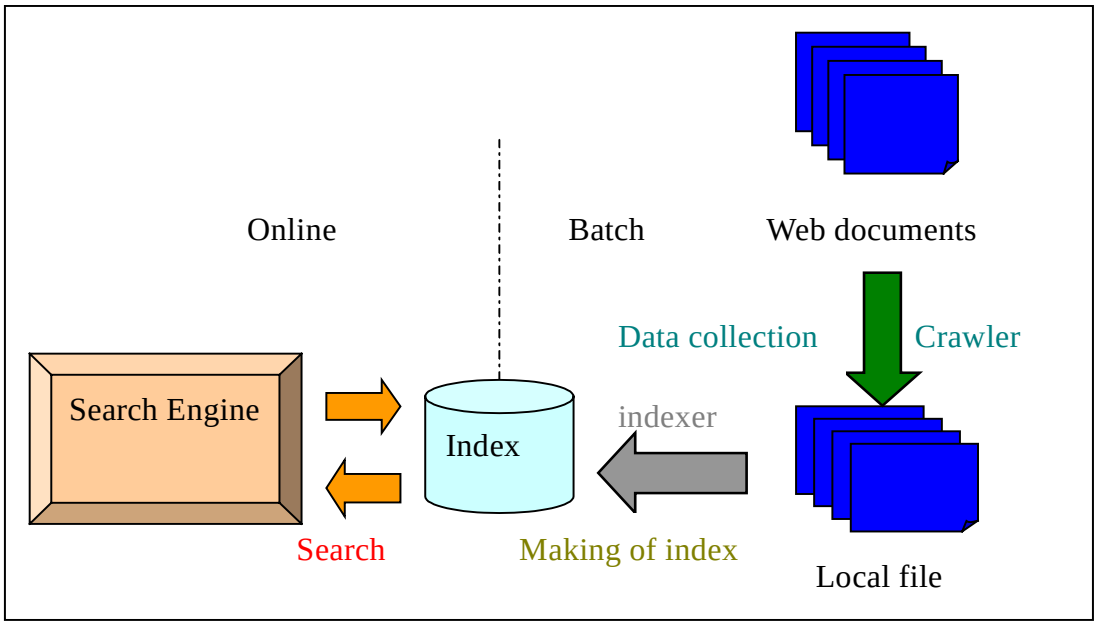

Fig 1 Management process of robot type search engine

2.1.1.2.1.Management process 1 - data collection

The crawler of the robot type search engine traces the hyperlink and goes it round Web. Moreover, the URL is visited when there is a request of the URL registration etc.Thus, documents are collected.

2.1.1.2.2.Management process 2 - index making

The Web document to which the round is received with the crawler of the robot type search engine is stored in the index by the indexer. This work is called "Index" and "Indexing", etc.

The technology to enable URL suited as a retrieval result to be returned high-speed and accurately as a retrieval result by converting into the data which treats easily for the search engine and doing the index to the data base.

At this time, which word is included in which document or the related index is made. However, because it is not easy for Japanese to cut out the word from sentences, the application of the distinct word cutting out is used.

The appearance frequency of the word is requested, the position

where the word appears is requested, the weight putting of each tag is applied, and the link information is used. The data which treats easily is made for the search engine by each search engine's using various becoming it technologies.

2.1.1.2.3. Management process 3 - search

The search engine listing displays URL in order with high retrieval key vs. and agreement level that the user input.

The adjacent level of the key word appearance position in the weight putting of the text match, the key word appearance frequency, and each tag and elements and the key word and clicks are popular, and considered various elements like the link popularity and the site theme, etc. by the retrieval algorithm of the program to which the robot type search engine requests the agreement level.

In the retrieval function offered by the robot type search engine, these elements are synthesized, the retrieval result scoring is done, and an appropriate retrieval result is derived.

2.1.2.Namazu

Namazu which is the representative of the Japanese full-text search engine is used for this thesis retrieval engine.

2.1.2.1.Feature of Namazu

Namazu is a full-text search engine intended for easy use. Not only does it work as a small or medium scale Web search engine, but also as a personal search system for email or other files.

Namazu is software to have aimed to construct the WWW full-text search system in the Konaka scale such as the unit of the Web server, units of Intranet, and specific field specialties easily. Development is done with Namazu Project. It corresponds to most OS now. The most standard Japanese full-text search software.

The program (following mknmz) which makes the index is described

with Perl. Retrieval command (namazu) is described by C. The time which hangs to the retrieval on the character of the algorithm is not influenced so much by the size of the index.

2.1.2.2.Management process of Namazu

2.1.2.2.1.Data collection

There is no crawler of Namazu. Then, GNUWget was used for substitution.

2.1.2.2.2.Index making

Kakasi [9]or chasen [10]can be used for cutting out of the word of Namazu. Chasen was used in this thesis. And, the weight putting of the tf/idf method and each tag is done to the method of the score.

2.1.2.2.3.search

As for the retrieval function of Namazu, various tool like the call from command line, CGI, and another application etc. is offered. However, to express the result in the sight, CGI is targeted in this thesis. In addition, pnamazu[11] is used for CGI client in this thesis.

2.1.2.3.Technology related to Namazu

2.1.2.3.1.GNU Wget

GNU Wget is a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive commandline tool, so it may easily be called from scripts, cron jobs, terminals without Xsupport, etc. The link is analyzed, and the file is acquired. It follows Robot.txt. The acquired file type is specified. The acquired domain can be specified.There is a feature.

2.1.2.3.2.chasen

ChaSen version 1.0 is officially released on 19 February 1997 by

Computational Linguistics Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology (NAIST). It is a FREE Japanese Morphological analyser. It grows out of developing JUMAN version 2.0 and has made a significant improvement in system performance.

2.1.2.3.3.The tf/idf method

The tf/idf method is a method of calculating the importance degree to the word in the document in consideration of the following two points. [12]

- Does the word appear at how much frequency in the document?
- How much does not the word appear with other documents?

The importance degree of the word is concretely calculated by the following methods. The document of N piece is included in index idx. Key word t is done. The number of documents including t is assumed in idx and the frequency of t df and in the document is assumed to be tf . Weight w of the key word in the document is defined as follows.

$$w = tf \times \log \frac{N}{tf}$$

The difference of each document clearly becomes a numerical value from the importance degree only of the appearance frequency of the word by using the tf/idf method and it appears. Therefore, the Case sensitive matching of the document is expected to improve.

2.1.2.3.4.pnamazu

Pnamazu is a retrieval client of the Perl version by Mr. Furukawa Rei. If the file named NMZ.format exists, the control of the format of CGI output can be facilitated. It is written with Perl. The feature is enumerated. An original format was made in this thesis.

2.1.3.SOM

2.1.3.1.About SOM

One though SOM (Self Organization Map) modeled the neural net work、

The feature of input data according to a certain distribution is extracted by the competition and reinforced study and the neighborhood study without S teachers, the distribution is generated, and the approximated feature map is generated Because many of maps are displayed in two dimension plane (There is the one of one dimension, too), data with a similar feature is output to a near position in the map. Therefore, it is easy to understand in the sight because it has a feature similar to which data is understood data's to which position of the map it having been output.

The network of SOM consists of two layers of the output layer where the output node was arranged like the lattice in the input layer and two dimensions, and the input layers unite with all the output nodes. The uniting vector which corresponds to input data is stored in each output node, and the uniting vectors of the output nodes in the output node with the input vector and the first uniting Kon vector and the neighborhood are brought close to the input vector. The area where data with a similar feature gradually gathers mutually is made from the repetition of such operation. That is, the map is self-organized, and the map which reflects the feature of the input vector is generated.

Fig 2 is the one that SOM of one road one capital two prefecture 43 prefectures of Japan was done by demographic data. There are 21 items of the number etc. of children as an input vector population by sex from 1985 all population to 1997 from those who move in, those who transfer it, and 1985 to 1997, areas, and children's ratios.

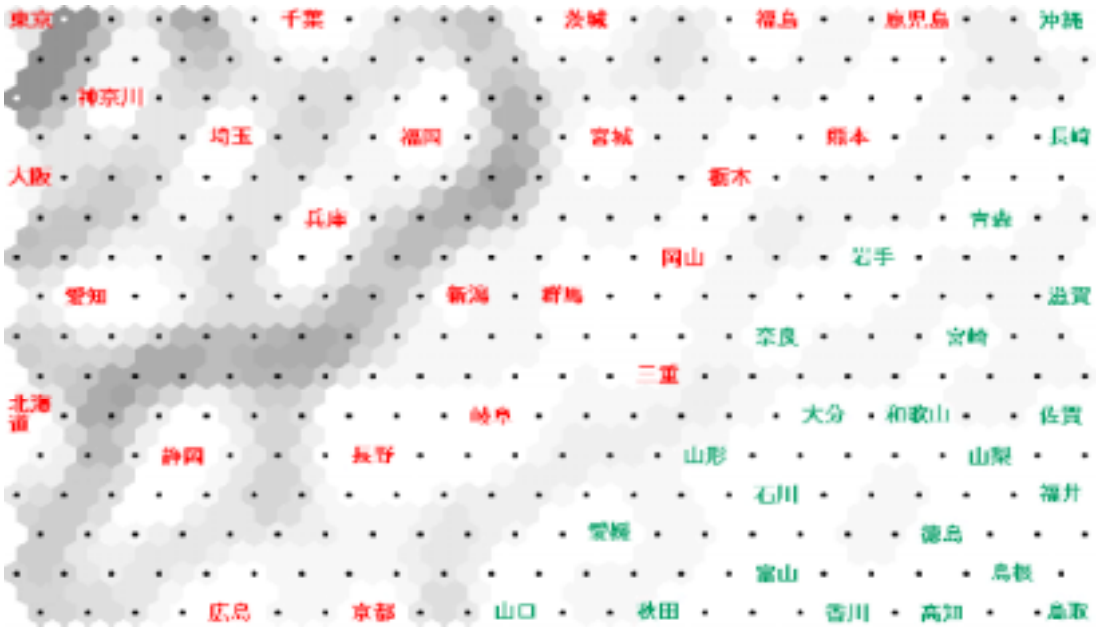

Fig 2 Application example to demographic data

2.1.3.2. Learning algorithm of SOM

The learning algorithm of SOM is as follows.

(1) A suitable value is set to uniting weight U_i which connects the input layer with a rival layer. U_i is given by the same dimension as input vector E.

(2) Input vector E is presented in the input layer.

(3) Agreement value $\|E-U_i\|$ of weight vector U_i to neuron i in input vector E and a rival layer is calculated. $\|\ \|$ shows the distance of the Euclid between vectors. This value measures the degree that the uniting weight of each neuron is corresponding to the value that the input vector corresponds.

(4) The neuron with the weight vector which resembles the input neuron to which the agreement value is minimized , that is, input vector E most is made a victor neuron. In addition, area N_c which is called a neighborhood area in surroundings of the victor neuron is set.

(5) The weight vector is updated in the next expression.

$$\Delta u_{ij} = \alpha\left(E_j - u_{ij}\right)_c \quad i \subseteq N_c \qquad (1)$$

Δu_{ij} : Amount of update of uniting weight u_{ij}

α : Study rate

(6) Processing from 2 to 5 is repeated for each input vector.

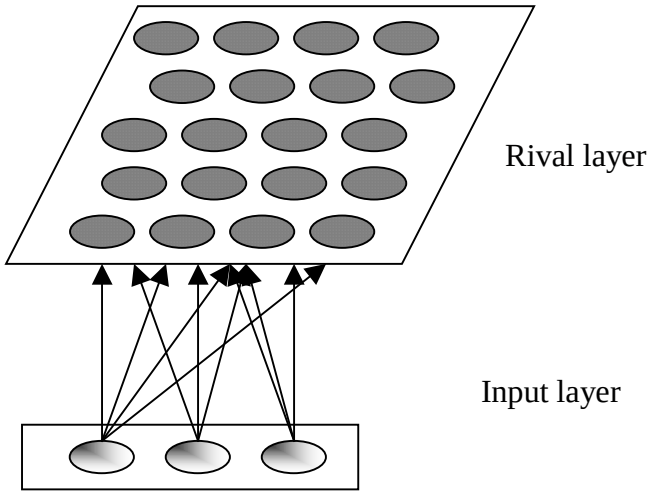

Fig 3 Network chart of SOM

The attenuation function of the neighborhood area and the study rate uses the next expression which transforms the Shigmoid function generally used by the BP method as an I/O response function.

$$N_c = \left[(N_{c\max} - N_{cmin}) \frac{f\left(t/t_{all}\right) - f(1)}{1 - f(1)} + N_{cmin} \right] \qquad (2)$$

$$\alpha = (\alpha_{\max} - \alpha_{min}) \frac{f\left(t/t_{all}\right) - f(1)}{1 - f(1)} + \alpha_{min} \qquad (3)$$

$$f(x) = \frac{1}{\exp(x)} \qquad (4)$$

t : Present study frequency

t_{all} : All study frequency which should do

$N_{c\max}$, $\alpha_{\max}$: Initial value of neighborhood area and study rate

N_{cmin} , α_{min} : Closing share price of neighborhood area and study rate

The neighborhood area and the study rate are attenuated like this, and only the specified frequency is studied repeatedly. And, it is output to the neuron in a rival layer where each input vector is the most corresponding to the uniting weight in the obtained map. A similar relation of each input vector can be expressed in the sight by this position.

2.1.4.BLSOM

2.1.4.1.What is BLSOM?

BLSOM (batch study type Self Organization map) is one of SOM. It was designed by Abe at National Institute for Genetics and Yamagata University.

SOM has a big influence on the map formed after it studies in the data input from the character of the learning algorithm back. In addition, because an initial value is suitably decided, the result might not settle. BLSOM is an improvement of those problems by doing the improvement and the principal component analysis on the algorithm of SOM.

Fig 4 is the one that SOM was done by the gene data. SYN and ECO of being necessary to become one area are two or more originally areas. It is thought that the reason for this is that an initial value was random. Then, the one that the same data was processed with BLSOM is Fig 5. The reason why the area size is different is that the influence power changes in SOM depending on the input order.

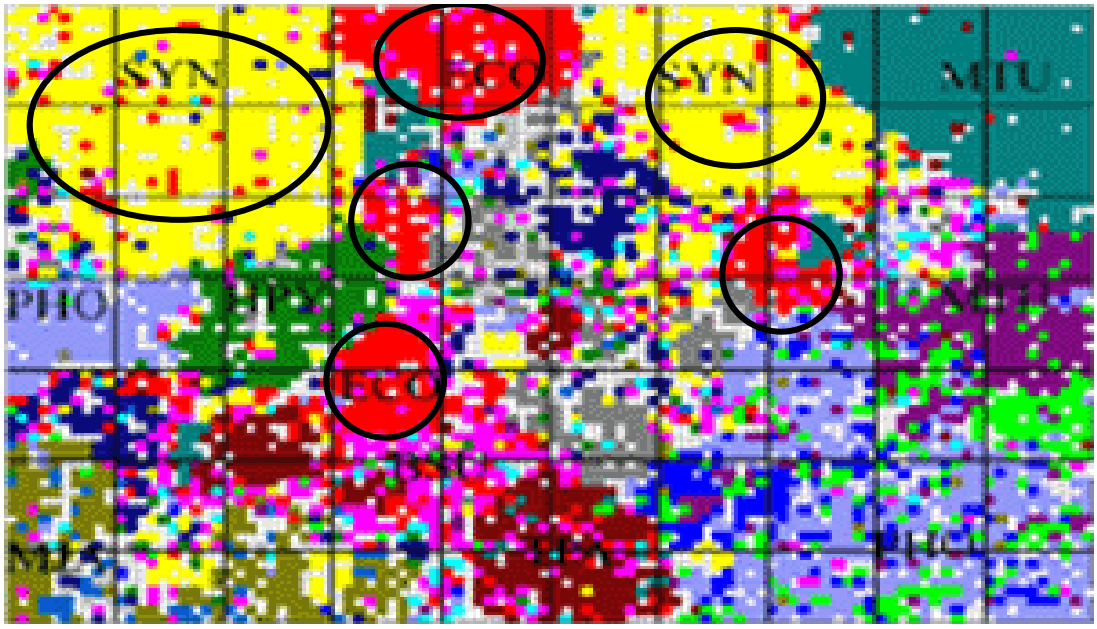

Fig 4 Result in SOM

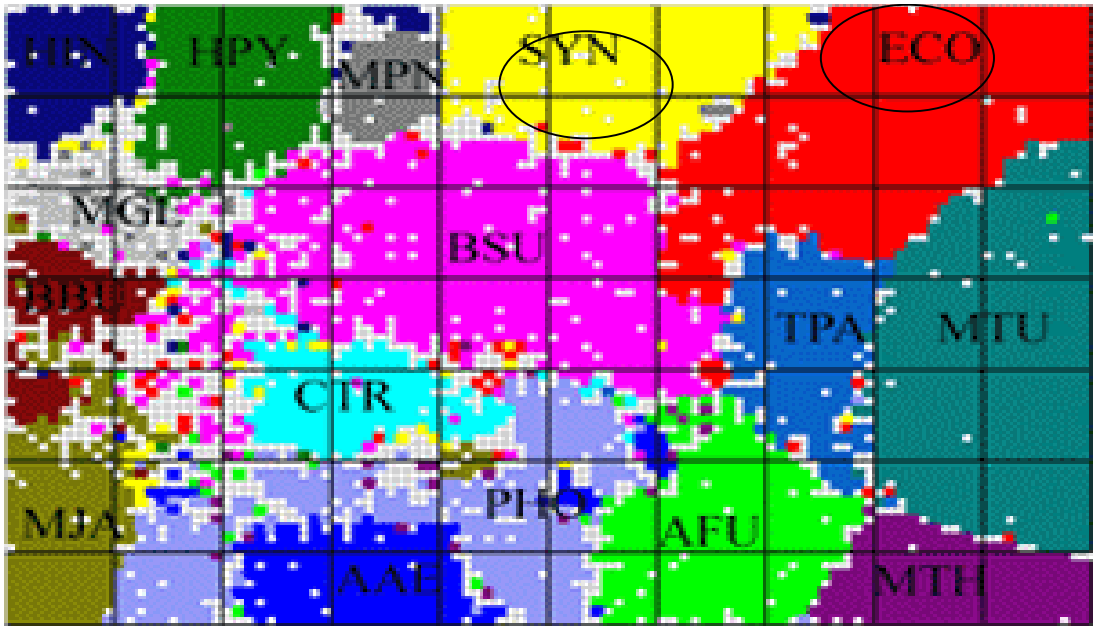

Fig 5 Result in BLSOM

2.1.4.2.Learning algorithm of BLSOM

The learning algorithm of BLSOM is the following. 。

(1) The principal ingredient is analyzed in the input layer.

(2) Uniting weight U_i which connects the input layer with a rival layer

is set by the following expressions. U_i is given by the same dimension as input vector E.

$$U_i = E_{avg} + 5\sigma_1\left(b_1 \frac{s + S/2}{S} + b_2 \frac{t + T/2}{T} \right) \qquad (5)$$

s, t, S, T:The 1st and two axes of BLSOM

σ_1 : Decentralized value of the first principal ingredient

E_{avg} : Mean vector of input layer

b_1, b_2 : The 1st and two principal ingredient vector

(3) Input vector E is presented in the input layer.

(4) Agreement value $\|E - U_i\|$ of weight vector U_i to neuron i in input vector E and a rival layer is calculated. $\| \|$ shows the distance of the Euclid between vectors.

(5) The neuron with the weight vector which resembles the input neuron to which the agreement value is minimized, that is, vector E most is made a victor neuron.

(6) The input layer is classified for the victor neuron in all input layer.

(7) Each weight vector is updated in the next expression.

$$U_i^{(new)} = U_i + \alpha(r)\left(\sum_{i=1}^{N_{st}} \frac{E_i}{N_{st}} - U_i \right) \qquad (6)$$

$\alpha(r) = \max\{0.01, 0.06(1 - r/R)\}$

r:Distance from victory neuron

R:Neighborhood distance

N_{st} : Number of input layers where U_i is made victory neuron

2.1.5. Principal component analysis method

2.1.5.1. What is principal component analysis method?

Two kinds of variables of 15 people like the height and weight, etc. are measured by the physical examination, and it is assumed that it becomes

like Fig 6 when this data is plotted.

The tendency that weight grows by growing of the height from this figure, too can be understood. It is likely to only have to think about the axis like z_1 newly in figure to express this tendency. It can be interpreted that this z_1 axis shows "Size of the body". However, even only "Size of the body" is not expressible of all characteristics of everybody. There is a person with thin person endowed with generous girth in 15 people, too. Then, it thinks about an orthogonal z_2 axis to the z_1 axis. It can be interpreted that this z_2 axis shows "Level of obesity". Thus, the variable "Height" and "Weight" is not independently treated. It comes to be able to understand the relation and the feature between variables included in data easily by introducing the overall index said, "Size of the body" and "Level of obesity". Such the overall index is statistically set, and the technique to understand the relations between variables is the one which is called that it is principal component analysis (PCA:Principal Component Analysis). The principal ingredient is the overall index. z_1 which shows "Size of the body" is called the first principal ingredient in the example of the physical examination. And, z_2 which shows "Level of obesity" is called the second principal ingredient.

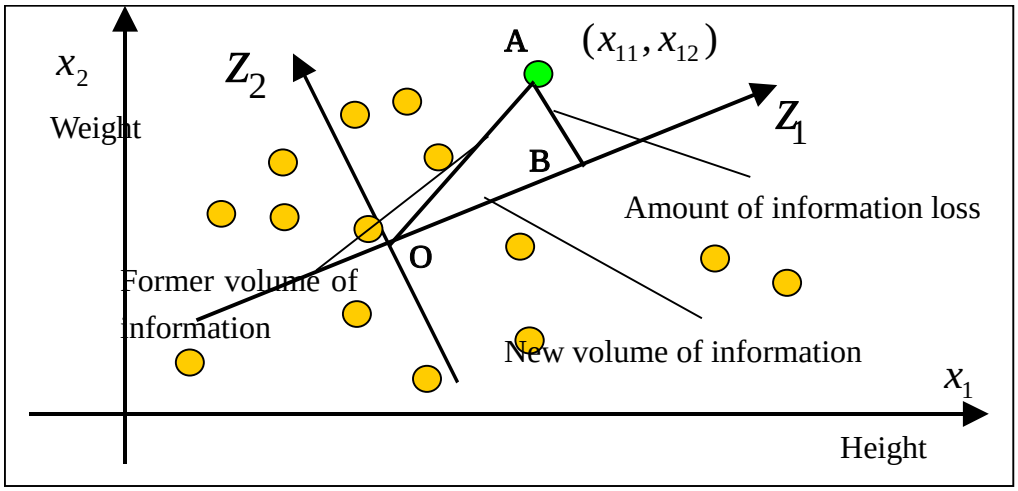

Fig 6 Result of physical examination

Because the feature of data (Or, tendency) is roughly expressible by z_1 which shows "Size of the body", data can be represented only by z_1. In

this case, the data expressed by two kinds of variables will be expressed by one kind of variable. That is, the data of two dimensions was converted into the data of one dimension while suppressing the loss of information to the minimum by showing the feature of data by the principal ingredient. It is possible to think the principal component analysis to be a technique to former make data low-level from such a viewpoint.

When the result of the physical examination is represented by "Size of the body", information on "Level of obesity" will be lost. The amount of the loss of information on that time is shown though the former making data low-level is accompanied by the loss of information like this by the perpendicular length sold wholesale from each data point to the 1st kind element z_1. It is necessary to reduce the amount of the information loss as much as possible to most often express the feature of data. Actually, the principal ingredient is decided under the condition of minimizing the amount of the information loss in the principal component analysis.

Minimizing the amount of the information loss. It is turning of the maximization of the obtained volume of information inside out. This can be achieved by setting the principal ingredient in the direction where the direction where the difference of data is shown most ,that is, decentralization becomes the maximum. Let's show this with Fig 6. Now, the amount of the information loss when data is represented only by the first principal ingredient z_1 becomes perpendicular length AB lowered from point A to straight line z_1 if paying attention to point A(x_{11}, x_{12}). z_1 is used and the obtained volume of information is $\overline{OB}$. Here, point 0 is a starting point of the z_1 axis, and the center of gravity of data. This new volume of information $\overline{OA}$ is called the first principal ingredient score. The relation named

$$OA^2 = OB^2 + AB^2 \qquad (7)$$

is obviously approved between an amount of the information loss and new volume of information (principal ingredient score). When $\overline{OA}$ will be called former volume of information, the relation named

(square harmony of former volume of information)=

(square harmony of new volume of information)+
(square harmony of amount of information loss) of the entire data of 15 points is approved.
Maximizing the square harmony of the amount of Shin paving Jou report taking because of a certain constant the square harmony of former volume of information here when the square harmony of the loss volume of information is minimized understands and it is understood that it is equivalent.

There is essentially no difference up to here for the multivariable about "Height" and "Weight" though it has described in case of two variables. That is, the principal component analysis is a technique for expressing information on variable $\{x_p\}$ $(p = 1,2,\Lambda, P)$ of P piece by using principal ingredient $\{z_m\}$

$$z_m = \sum_{p=1}^{P} w_{pm} x_p \qquad (m = 1,2,\text{☹}, M) \qquad (8)$$

of M$(M \le P)$piece independent given as a linear combination of $\{x_p\}$while suppressing the loss of information to the minimum each other (overall index). z_m is called m principal ingredient, and the coupling factor $\{w_{pm}\}$ $(p = 1,2,\Lambda, P : m = 1,2,\Lambda, M)$ is decided to satisfy the following conditions.

< condition >

The first principal ingredient z_1 decentralization is the maximum in the decentralization of all the first types of $\{x_p\}(p = 1,2,\Lambda, P)$, and m principal ingredient $\{z_m\}(m = 2,3,\Lambda, M)$ decentralization is maximum in the decentralization of 1 next expressions no correlation to all $\{z_{m'}\}(m = 2,3,\Lambda\ M)$. However, it is assumed

$$\sum_{p=1}^{P} w_{pm}^2 = 1 \qquad (m = 1,2,\text{☹}, M) \qquad (\mathbf{9})$$

2.1.5.2.Deriving of principal ingredient

2.1.5.2.1.Preparation

The method of deciding the principal ingredient so that the decentralization of the principal ingredient may become the maximum according to the condition of describing in 2.1.5.1 is described. It thinks about the case with the sample of N piece about the variable of P piece now, and measurements are assumed to be $\{x'_{np}\}(n=1,2,\Lambda\ ,N:p\ =\ 1,2,\Lambda\ ,P)$. To do the following discussions easily, deflection $\{x_{np}\}$ of each variable from the mean value $\{\bar{z}_p\}(p=1,2,\Lambda\ ,P)$ is introduced. That is, it is assumed

$$x_{np}=x^*_{np}-\bar{x}_p(n=1,2,\text{☹},N;p=1,2,\text{☹},P). \qquad (10)$$

At this time, the entire measurement data is given for the following procession X.

$$X=\begin{pmatrix} x_{11} & x_{12} & \text{☹} & x_{1P} \\ x_{21} & x_{22} & \text{☹} & x_{2P} \\ \text{💣} & \text{💣} & \text{⚑} & \text{💣} \\ x_{N1} & x_{N2} & \text{☹} & x_{NP} \end{pmatrix} \qquad (11)$$

2.1.5.2.2. Deriving of the first principal ingredient

Because the first principal ingredient z_p is given by expression (8), the coupling factor is assumed to be .

$$w_1=\begin{pmatrix} w_{11} \\ w_{12} \\ \text{💣} \\ w_{P1} \end{pmatrix} \qquad (12)$$

.Value x_{n1} of the first principal ingredient z_1 corresponding to nth sample

$$x_n=(x_{n1}\quad x_{n2}\quad \text{☹}\quad x_{nP}) \qquad (13)$$

becomes

$$\begin{aligned} t_{n1}&=\sum_{p=1}^{P}w_{p1}x_{np} \\ &=x_nw_1 \end{aligned} \qquad (14)$$

Value x_{n1} of this first principal ingredient z_1 is called the first principal ingredient score. The first principal ingredient score corresponding to the sample of N piece is brought together in one vector, and it puts it with

$$t_1 = \begin{pmatrix} t_{11} \\ t_{12} \\ \bullet \\ t_{N1} \end{pmatrix} \qquad (15)$$

Then,

$$t_1 = Xw_1 \qquad (16)$$

is approved. Average $\overline{t_1}$ of the first principal ingredient score is

$$\begin{aligned} \overline{t_1} &= \frac{1}{N}\sum_{n=1}^{N} t_{n1} \\ &= \frac{1}{N}\sum_{n=1}^{N} x_n w_1 \\ &= \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{p=1}^{P} w_{p1} x_{np}\right) \\ &= \frac{1}{N}\sum_{p=1}^{P} w_{p1}\sum_{n=1}^{N} x_{np} \\ &= 0 \end{aligned} \qquad (17)$$

The first principal ingredient z_1 decentralization $\sigma_{z_1}^2$ becomes

$$\begin{aligned} \sigma_{z_1}^2 &= \frac{1}{N-1} t_1^T t_1 \\ &= \frac{1}{N-1}(Xw_1)^T (Xw_1) \\ &= w_1^T V w_1 \\ &\geq 0 \end{aligned} \qquad (18)$$

Procession V is a nonnegative fixed value procession who is called a

covariance procession, and it is given by

$$V = \frac{1}{N-1} X^T X \qquad (19)$$

As for the (i, j) element v_{ij} it is

$$v_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} x_{ni} x_{nj} \qquad (20)$$

$$= \frac{1}{N-1} \sum_{n=1}^{N} \left(x_{ni}^{*} - \bar{x}_i \right) \left(x_{nj}^{*} - \bar{x}_j \right) \qquad (21)$$

Moreover, $v_{ij} = v_{ji}$, that is, $V = V^T$ is obviously approved.

The first principal ingredient z_1 should be decided for the decentralization $\sigma_{z_1}^2$ to become the maximum under expression (9). This optimization problem can be easily solved by using the Lagrange multiplier method. Lagrange multiplier λ is introduced, and that is, coupling factor w_1 which maximizes J_1 only has to be put, and requested

$$J_1 = w_1^T V w_1 - \lambda \left(w_1^T w_1 - 1 \right) \qquad (22)$$

Then, J_1 is done by w_1 during the Catayobi minute with 0 when putting it. and it becomes

$$\frac{\partial J_1}{\partial w_1} = \begin{pmatrix} \frac{\partial J_1}{\partial w_{11}} \\ \frac{\partial J_1}{\partial w_{21}} \\ \vdots \\ \frac{\partial J_1}{\partial w_{P1}} \end{pmatrix}$$

$$
\begin{aligned}
&= \begin{pmatrix} 2\sum_{p=1}^{P} v_{1p} w_{p1} \\ 2\sum_{p=1}^{P} v_{2p} w_{p2} \\ \vdots \\ 2\sum_{p=1}^{P} v_{Pp} w_{pP} \end{pmatrix} - 2\lambda \begin{pmatrix} w_{11} \\ w_{21} \\ \vdots \\ w_{P1} \end{pmatrix} \\
&= 2Vw_1 - 2\lambda w_1 \\
&= 0 \qquad (23)
\end{aligned}
$$

Therefore, the conditional expression named

$$(V - \lambda I)w_1 = 0 \qquad (24)$$

is obtained. It does not become an eigenvalue problem , and Lagrange multiplier λ can show the requirement which should be met by using eigen equation

$$\det|V - \lambda I| = 0 \qquad (25)$$

this. Therefore, it is understood that coupling factor w_1 of Lagrange multiplier λ and the first principal ingredient z_1 is given respectively as an eigenvalue and an eigenvector of covariance procession V.

Well, it is likely to correspond to coupling factor w_1 that which eigenvalue in that maximizes the first principal ingredient z_1 decentralization $\sigma_{z_1}^2$ (eigenvector) though P piece exists in the eigenvalue λ because covariance procession V is P positive dimension procession. It is necessary to be given the first principal ingredient z_1 decentralization $\sigma_{z_1}^2$ by expression (20), and satisfy condition (25) by the coupling factor w_1. Then, when (25) is substituted for (20), and $w_1^T w_1 = 1$ is noted,

$$
\begin{aligned}
\sigma_{z_1}^2 &= w_1^T V w_1 \\
&= w_1^T \lambda w_1 \\
&= \lambda \qquad (26)
\end{aligned}
$$

is obtained. It is understood that the first principal ingredient z_1 decentralization $\sigma_{z_1}^2$ is more equal to eigenvalue λ of covariance procession V. Therefore, decentralization $\sigma_{z_1}^2$ of the first principal ingredient z_1 which should be maximized becomes equal to the maximum eigenvalue of covariance procession V, and the coupling factor w_1 can be requested as an eigenvector corresponding to the maximum eigenvalue.

2.1.5.2.3. Deriving of m principal ingredient

It is possible to request it according to the procedure that coupling factor $\{w_m\}(m = 2,3,\Lambda\ M)$ below the second principal ingredient is similar to the first principal ingredient. However, it is necessary to decide coupling factor w_m to become the maximum in the decentralization of 1 next expressions <no m principal ingredient $\{z_m\}(m = 2,3,\Lambda\ M)$ decentralization SZM correlation to all principal ingredient $\{z_{m'}\}(m' = 2,3,\Lambda\ m-1)$>. It is necessary to fill expression (9) naturally.

Here, the method of deriving coupling factor w_m of m principal ingredient is shown by using the principle of induction. Even the m-1 principal ingredient is requested, and those coupling factors $\{w_i\}(i = 2,3,\Lambda\ m-1)$ are assumed to meet requirement

$$(V - \lambda_i I)w_i = 0 \qquad (27)$$

$$w_i^T w_i = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases} \qquad (28)$$

And, coupling factor w_m of m principal ingredient is requested by using the Lagrange multiplier method. That is,

$$J_m = w_m^T V w_m - \lambda_m (w_m^T w_m - 1) - \sum_{i=1}^{m-1} \mu_i w_m^T w_i \qquad (29)$$

and w_m which puts and maximizes J_m are obtained. Here, a right final paragraph shows the condition that all of m principal ingredient $\{z_m\}(m = 2,3,\Lambda\ M)$ and principal ingredient $\{z_{m'}\}(m' = 2,3,\Lambda\ m-1)$ become no correlations. J_m is done by w_m during the Catayobi minute and it puts it

with 0. Then, it becomes

$$\frac{\partial J_m}{\partial w_m} = 2Vw_m - 2\lambda_m w_m - \sum_{i=1}^{m-1}\mu_i w_i$$
$$= 0 \qquad (30)$$

When $w_j^T(j=1,2,\cdots,m-1)$ is put from the left, and expression (30) is used,

$$w_j^T V w_m - \mu_j = 0 \quad (j=1,2,\cdots,m-1) \qquad (31)$$

A=B is obtained. Here, clause 1 : from $V=V^T$ and expression (29).

$$w_j^T V w_m = w_m^T V w_j$$
$$= w_m^T \lambda_j w_j$$
$$= 0 \quad (j=1,2,\cdots m-1) \qquad (32)$$

Expression (32) because of its becoming it

$$\mu_j = 0 \quad (j=1,2,\cdots,m-1) \qquad (33)$$

It becomes it. When this result is substituted for expression (31),

$$(V-\lambda_m I)w_m = 0 \qquad (34)$$

is finally obtained. This is the same expression as expression (29), and it is understood that m principal ingredient z_m decentralization $\sigma_{z_m}^2$ is also equal to the eigenvalue of covariance procession *V*. However, because the eigenvector corresponding to the eigenvalue of m-1 piece and it has already been used from large one to show the principal ingredient to m-1, m principal ingredient z_m decentralization $\sigma_{z_m}^2$ becomes equal to a large eigenvalue, and can request coupling factor w_m to m turn eyes of covariance procession V as an eigenvector corresponding to the eigenvalue.

2.1.5.3. Standardization of data

Method to decide the principal ingredient based on the covariance V was described here by using measurements of each variable as it was. However, a principal ingredient different according to how to take the unit will be obtained when measured by the unit with a different each variable.

Moreover, if the principal component analysis is applied to the variable that it is large even if the unit is the same and decentralization is different as it is, a big variable to distribute will strongly influence the result. The relations between variables will be not able to be understood correctly. Therefore, it is necessary to standardize all variables by using some methods.

It is a standardization method of one on the average the method used most easily and widely as for each variable like becoming <decentralization> one. Concretely,

$$\tilde{x}_{np} = \frac{x_{np}^{*} - \overline{x}_{p}}{\sigma_{x_p}} \qquad (35)$$

A=B is used instead of using the measurements $x_{np}^{*}(n=1,2,\cdots N; p=1,2,\cdots P)$ when there is a sample of N piece about the variable of P piece. Here, $\overline{x}_{p}, \sigma_{x_p}^{2}$ is a mean value and standard deviation of variables x_{p} of p turn eyes respectively. In this case, because the covariance between standardized variables becomes equal to the correlation coefficient, the correlation matrix of data procession $\overline{X}$ to whom it is standardized is assumed to be

$$R = \frac{1}{N-1}\overline{X}^{T}\overline{X} \qquad (36)$$

Then, m principal ingredient z_m decentralization $\sigma_{z_m}^{2}$ becomes equal to a big eigenvalue, and can request coupling factor w_m to m turn eyes of correlation matrix R as an eigenvector corresponding to the eigenvalue by m principal ingredient z_m.

Doing the analysis which does not depend on the unit becomes possible by giving the standardization of the above-mentioned. However, actual measurement data is led and the analysis which uses the variable to which the influence of the error margin has been strongly received might lead a wrong result because it contains error margin the size and the character different and a variety of. Then, the method of applying different weight to each variable according to the level of the influence of the error margin is devised. As how to put weight though variety is thought, Error

margin, influence, at all, receive, variable, all, decentralization, become, error margin, change, the, decentralization, become, have, intuitive, appropriate.There is a method of using

$$\tilde{x}'_{np} = \frac{x^*_{np} - \bar{x}_p}{\sigma_{x_p}} \frac{\sigma_{x_p} - \sigma_{x_{pe}}}{\sigma_{x_p}} \qquad (37)$$

as how to put such weight instead of measurements $x^*_{np}(n = 1,2,\text{☹}\, N; p = 1,2,\text{☹}\, P)$. Here, $\sigma_{x_{pe}}$ is standard deviation of the error margin included in variables x_p of p turn eyes. Actually, accurately understanding the standard deviation of the error margin is difficult, and when the presumption value is given by some methods, becomes injuring effective with the weight described here.

2.1.6.Virtual Reality Modeling Language

After Virtual Reality Modeling Language(VRML) is a structurizing language in the virtual reality design language which describes virtual reality three dimension geometry, and VRML1.0 is announced in the autumn of 1994, the improvement has been piled up. VRML2.0 announced afresh develops, and and, standardization is advanced by International Organization for Standardization ISO(International Organization for Standardization) and international electronic engineering committee IEC(International Electrotechnical Commission), the specification is fixed as standard ISO/IEC14772 international on August 4, 1996, and the browser development manufacturer and the vender are developing VRML a browser.

2.1.6.1.Feature of VRML

It characterizes, and it is enumerated that it is the following of VRML. 。

- It is a language which describes interactive 3 Dimension graphics. VRML is Web aim language for three dimension computer graphics (3DCG). This language differs from a usual programming language, does the form of an interactive model description language, and can construct three dimension virtual world easily.

- It corresponds to the multimedia. VRML corresponds to the multimedia, and can open music, the photograph, and the video image to the public in the virtual world.
- The construction of the virtual world combines objects of basic shape (object), and creates a more complex object.
- It is also possible to put on the homepage and to open the constructed virtual world to the public all over the world. If this homepage is read by Web a browser, it displays in the browser and the VRML viewer that the plug-in is done displays the content of the VRML file on the screen as CG.
- Because the VRML file is described by the text, volume of data is small.

2.2.Relation research

Clustering of the retrieval result is enumerated as a relation research.

2.2.1.vivisimo

It is a meaning "It is wise" in vivisimo Spanish. Retrieving it at a dash by the use of two or more sites by the meta search engine developed by the team of Carnegie Melon University can also specify the site and this be retrieved. It is likely to have a hard time though there are a lot of retrieval results and a target page is found when the meta retrieval is put. However, it is easy for a target document to look for the direct retrieval because it classifies it in vivisimo according to the content.

The relation to this thesis is to classify it automatically like being easily searchable from the retrieval result. However, it is assumed the map display by VRML in the thesis though vivisimo is output by the list after the retrieval result is classified.

Chapter3. Mounting and evaluation of system

3.1. Outline of this system

This system is a composition as shown in Fig 7.

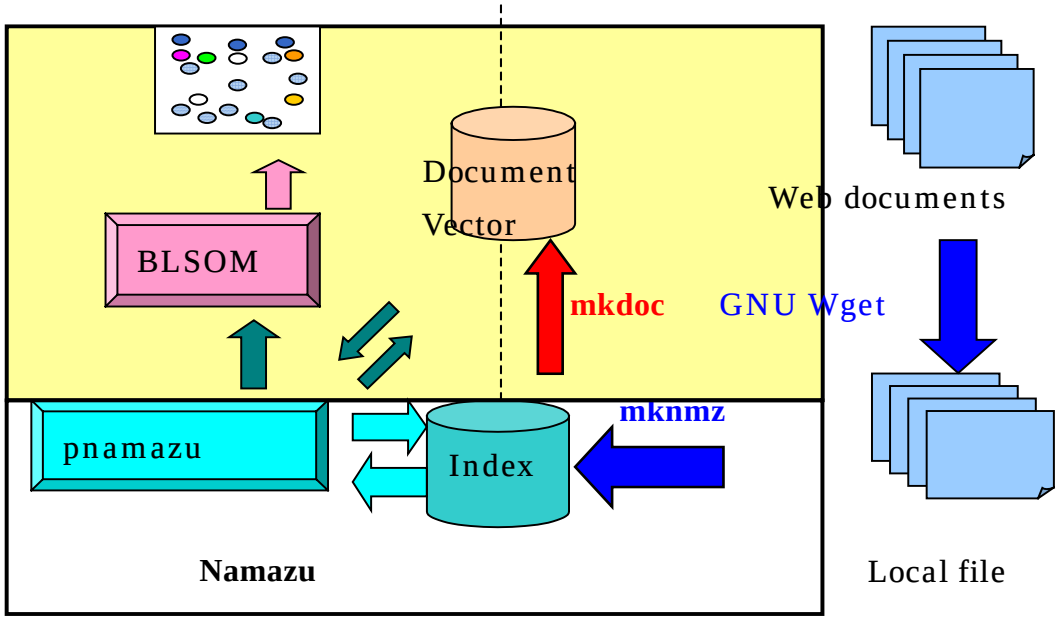

Fig 7 System Configuration

Data generation

- Data is collected by using GNU Wget.
- Index file (NMZ.*) is made by using mknmz with collected files.
- The file of the vector of the document is made from the index file by the following algorithms.

Retrieval of data

- The user inputs the retrieval key to the form of CGI.
- Pnamazu processes and the document group which matches it to the retrieval key is acquired. 。
- The vector of the document of each document is read, and the vector of the document is acquired.
- BLSOM is done by the vector of the document.

■ The result of BLSOM is displayed with VRML.

3.2.Details of system mounting

3.2.1.Data making part

3.2.1.1.Data collection

The following options were specified for $HOME/.wgetrc beforehand in this thesis though various options were able to be specified for GNUWget.

tries = 2

reclevel = 0

waitretry = 2

timestamping = on

robots = off

accept=html,HTML,htm,HTM,txt,TXT,shtml,SHTML,shtm,SHTM,xml,XML,xhtml,XHTML,xhm,XHTM

Which kind of files are collected whether to collect robot.txt is sequentially specified on at depth in which the frequency and the hierarchy of Ritorai are traced and the weight time. This is to collect even the deepest points though depth in which the hierarchy is traced here is 0. The following option was used besides these options.

-D www.jaist.ac.jp

The option of the host specification was specified. The start of the data collection http://www.jaist.ac.jp/index.html.

3.2.1.2.Index making

Mknmz which was the program of namazu to the index making was used. Various options this program can be specified. The option used with this thesis is as follows.

-a

-U

--replace=s#$HOME/jaistdata/#http://

-O $HOME/jaistindex

All data is sequentially targeted on. Encode of URI is not done. The code to substitute URI is specified. The result is output to another directory.

Consequently, the following index files and the output control files, etc. are made.

NMZ.i	Index file
NMZ.ii	Index file for seek
NMZ.w	Word index file
NMZ.wi	Word index file for seek
NMZ.p	Index file for Fraz retrieval
NMZ.pi	Frazindeccsfail for seek

Table 1 Index file for Namazu

NMZ.field.date	Date information
NMZ.field.date.i	Information index of date
NMZ.field.from	Author
NMZ.field.from.i	Author index
NMZ.field.message-id	Message ID
NMZ.field.message-id.i	Message ID index
NMZ.field.newsgroups	Newsgroup
NMZ.field.newsgroups.i	Newsgroup index
NMZ.field.size	Size of file
NMZ.field.size.i	Index of size of file
NMZ.field.subject	Title
NMZ.field.subject.i	Title index
NMZ.field.summary	Summary
NMZ.field.summary.i	Summary index
NMZ.field.to	to
NMZ.field.to.i	To index
NMZ.field.uri	URI
NMZ.field.uri.i	URI index

Table 2 Document index file for Namazu

NMZ.body	Main body of the first screen for CGI
NMZ.body.es	Main body of the first screen for CGI(English)
NMZ.body.fr	Main body of the first screen for CGI(French)

NMZ.body.ja	Main body of the first screen for CGI(Japanese)
NMZ.foot	The lower side of screen for CGI
NMZ.foot.es	The lower side of screen for CGI(English)
NMZ.foot.fr	The lower side of screen for CGI(French)
NMZ.foot.ja	The lower side of screen for CGI(Japanese)
NMZ.head	The upper part of screen for CGI
NMZ.head.es	The upper part of screen for CGI(English)
NMZ.head.fr	The upper part of screen for CGI(French)
NMZ.head.ja	The upper part of screen for CGI(Japanese)
NMZ.tips	Retrieval tips for CGI
NMZ.tips.es	Retrieval tips for CGI(English)
NMZ.tips.fr	Retrieval tips for CGI(French)
NMZ.tips.ja	Retrieval tips for CGI(Japanese)
NMZ.result.normal	Result output layout
NMZ.result.normal.es	Result output layout(English)
NMZ.result.normal.fr	Result output layout(French)
NMZ.result.normal.ja	Result output layout(Japanese)
NMZ.result.short	Result output layout shortening version
NMZ.result.short.es	Result output layout shortening version(English)
NMZ.result.short.fr	Result output layout shortening version(French)
NMZ.result.short.ja	Result output layout shortening version(Japanese)

Table 3 Output control file for Namazu

NMZ.err	Error record when executing it
NMZ.version	Version of Namazu
NMZ.slog	Retrieval log
NMZ.t	The time stamp and the missing number for the document are recorded.
NMZ.r	File list in index
NMZ.log	Log when executing it

Table 4 Others

3.2.1.3.Making of document vector

The document vector is made by using NMZ.i made by mknmz. The data structure of NMZ.i is

[Word1¥n]

[Total of entry * 2][Document ID][Score][Document ID][Score]....¥n

[Word2¥n]
[Total of entry * 2][Document ID][Score][Document ID][Score]....¥n
[Word3¥n]
[Total of entry * 2][Document ID][Score][Document ID][Score]....¥n

.....

Then, it only has sequentially to read data from the beginning, and output the file with the following data structures oppositely. Word n is a numerical value here, and it is in NMZ.w an actual word. Therefore, word n is assumed to be word ID in the following.

[Document 1¥n]
[Word ID][Score][Word ID][Score]....¥n
[Document 2¥n]
[Word ID][Score][Word ID][Score]....¥n
[Document 3¥n]
[Word ID][Score][Word ID][Score]....¥n
.....

By the way, the data structure of NMZ.field.date is as follows.

Sat, 22 Jun 2002 00:00:00
Wed, 10 Oct 2001 00:00:00
Tue, 15 Jan 2002 00:00:00
Sat, 22 Jun 2002 00:00:00
Thu, 14 Mar 2002 00:00:00

Document ID does not exist in the file. This is not only NMZ.field.data. All NMZ.field.* is possible to say to the file. The data structure is changed in consideration of this as follows.

[Word ID,Score][Word ID,Score]....¥n
[Word ID,Score][Word ID,Score]....¥n
[Word ID,Score][Word ID,Score]....¥n
.....

And, the result is written in NMZ.field.vector.

In addition, the data structure of the score total of each document does

as follows.

Score total¥n

Score total¥n

Score total¥n

……

The result is written in NMZ.field.score.

The logic of mkdocv is shown in the figure, it becomes the following.

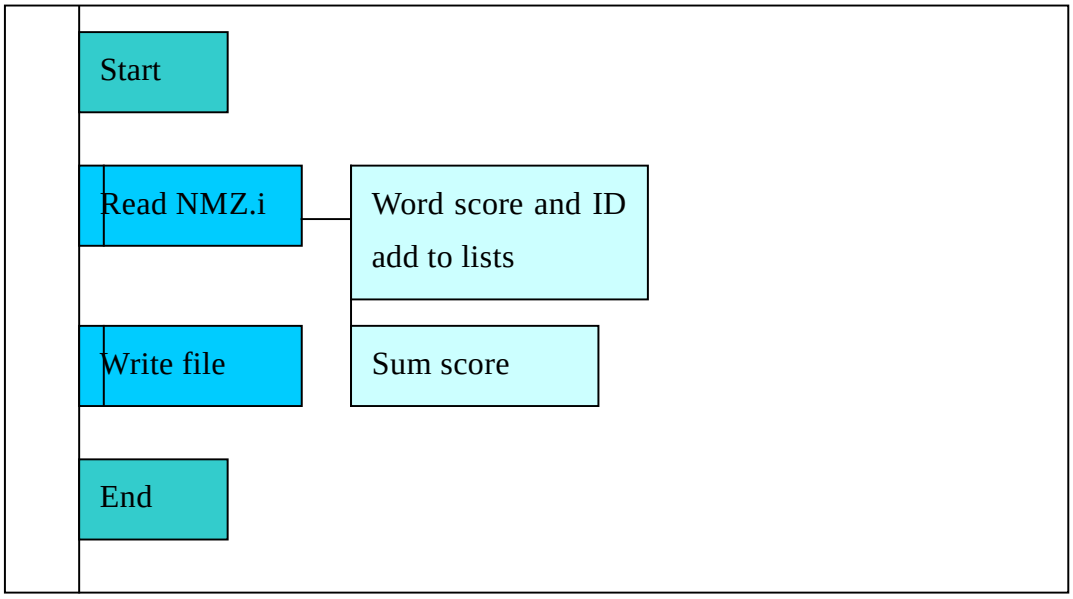

Fig 8 PAD of mkdocv

3.2.2.The retrieval of data is detailed.

3.2.2.1.Input of retrieval key

The user inputs the retrieval key to CGI.

3.2.2.2.Processing of pnamazu

Document ID that pnamazu outputs based on the retrieval key and the order of sorting is acquired. In addition, when the file named NMZ.format exists, pnamazu can control the output of CGI with the format file. Table 5, Table 6, and Table 7 are the commands, and variables can be used in the format file.

#word	The number of reference hits (number of hits of each words) is output.
#result	The retrieval result is output.
#if var	If variable var is the truth, it is output to # else or # endif.
#else	#The condition of if is reversed.
#endif	#The control by if is shut.
#pageloop	#Only the number of pages repeats the line to endloop.
#endloop	
#include FILE	FILE is read.
#eval string	The result of doing the word which substitutes the following variable and the character string in eval of string is output.

Table 5 Commands can be used in NMZ.format

page_index	Whether there are two or more pages or not?
Prev	Whether there is "Previous page" or not?
prev_url	URL on the previous page
prev_href	Tag including URL on the previous page<A HREF=...>
Next	Whether there is "Subsequent page" or not?
next_url	URL of subsequent page
next_href	Tag including URL of subsequent page<A HREF=...>
Reference	Whether the number of reference hits is displayed or not?
Whence	The first number number under display
Whither	The last number number under display
Hit	Number of hits
Key	Retrieval type
Quotekey	It is the one that quotemeta was done as for ${key}.
start_time	Retrieval beginning time
time_to_search	Time which hung to retrieval(second)
Current_time	Time when this variable is output
time_to_current	Time until this variable is output(second)
Lang	Language of output(en/ja)
Code	Japanese code(EUC-JP/Shift_JIS/ISO-2022-JP)

Table 6 Variable can be used in NMZ.format

page_number	Present page number
page_url	URL to present page
page_href	Tag including URL on present page<a href=...>
page_current	Whether it agrees to the page number a present page number's displaying it or not?

Table 7 Variable which can be used in page loop of NMZ.format

The instruction "BLSOM" is added in NMZ.format for the BLSOM output.

3.2.2.3. Acquiring the vector of the document.

Necessary information in each document spools by using document ID. Necessary information is a document number, URL, Title, a score total, and a document vector for the output. By the way, the word and the score are put as for the vector of the document. However, the range of the score is uneven by the document. It divides each total score, and the value from 0 to 1 is used.

Which word is used when the spool processing ends is decided. When the word decided the document is included, the score of the word has been decided is output. When the decided word is not included, 0 is output to the file.

3.2.2.4.Doing BLSOM.

BLSOM is done by using the result of 3.2.2.3. The study frequency is 500 times. The neighborhood radius is 3. The size of the axis of the principal ingredient is halves of the output number.

In x axis of the result of BLSOM, it is a principal ingredient and y axis is the second element. VRML can control the color. Then, the color was applied by using the principal component analysis for the object. R, G, and B show the first principal ingredient, the second element, and the third element. The range of the value sets the value as becoming $0/256 \le R,G,B \le 256/256$.

3.2.2.5.It displays it with VRML.

The VRML file is made by using the document number for the output requested by position, color, obtained by 3.2.2.4and 3.2.2.2.

3.2.3.Retrieval result

The result of doing the retrieval of data is actually Fig 9. The retrieval key is "data mining. "The output number is 50. The order of sorting is the score. The neighborhood distance is 3. The study frequency is 500 times.

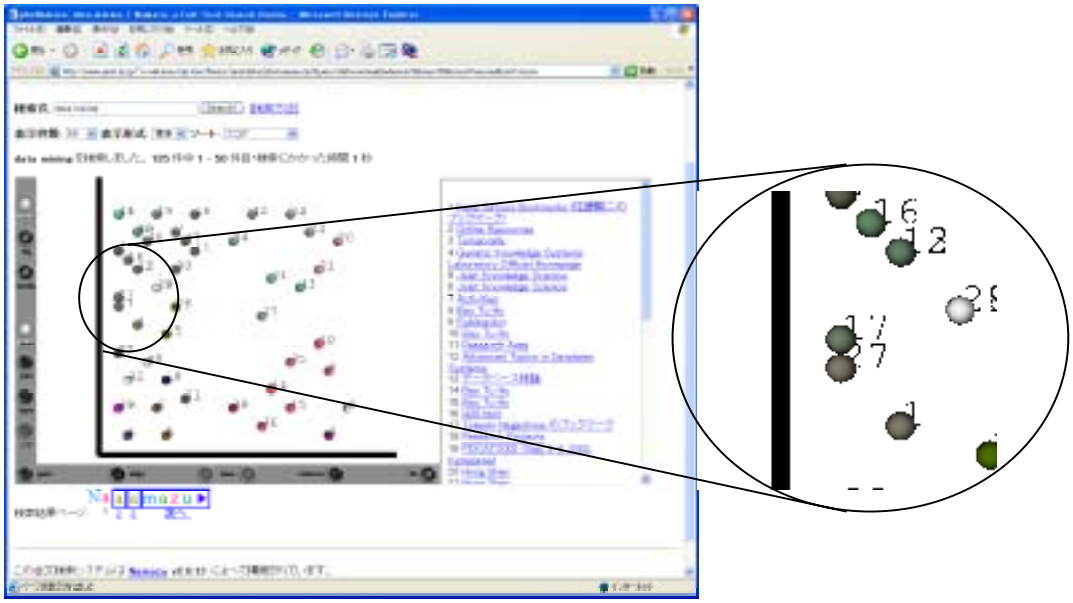

Fig 9 Result screen

No. 17 and No. 27 are adjacent. Both No. 17 and No. 27 were bookmark. However, it is complete others bookmark. The number shows the order. In a word , ten is away from rank.

3.3. Evaluation

The processing for the thesis consists of the data making part and the data retrieval part. The system evaluation also went separately respectively. The evaluation by the data making used the operation time of the program. The relevance ratio and the recall ratio were used for the evaluation of the retrieval of data.

3.3.1. Evaluation of data making

As for the data making, mkdocv of the program of making of the vector of the document was evaluated. mkdocv is a program makes NMZ.field.vector and NMZ.field.score reading NMZ.i. Because it is a program of the format

conversion alone, it is not significant to the evaluation of the result. Then, the evaluation was assumed to be an evaluation by comparing the time had been assumed to be necessary with mknmz. Timex was used for the time measurement. The result is Table 8.mkdocv is far, less the time required, and has ended because the algorithm is considerably easy.

	mknmz	mkdocv
real	2:56:22.41	18:35.42
user	1:35:46.84	17:55.94
sys	36:08.61	35.91

Table 8 Comparison at time of mknmz and mkdocv

3.3.2.Evaluation of retrieval of data

3.3.2.1.Precision ratio and recall ratio

The concept of ″ Relevance ″ which shows whether the retrieved document has suited the information demand to evaluate the information retrieval system is introduced. As for adaptability, the standard of the evaluation of recall ratio in which precision ratio which shows few of the retrieval noise and few of the retrieval leakage are shown is used.

When the judgment of adaptability is ″Suit″ or it is given no ″by binary of″, it is defined as

$$\text{Precision} = \frac{\text{Retrieved number of acceptable documents}}{\text{Retrieved number of documents}}$$

$$\text{Recall} = \frac{\text{Retrieved number of documents}}{\text{Number of all acceptable documents}}$$

However, expressions the following though the precision ratio and the recall ratio were shown were used. The reason is that it is not an evaluation of the search engine this time.

$k_i(1 \le i \le M)$: Word composes the group.

similar(k_i):Number of document has k_i

neighbor(k_i):Number of documents in group has k_i

$$\text{Precision} = \frac{1}{M}\sum_{i=1}^{M}\frac{(neighbor(k_i)\cap similar(k_i))}{neighbor(k_i)}$$

$$\text{Recall} = \frac{1}{M}\sum_{i=1}^{M}\frac{(neighbor(k_i)\cap similar(k_i))}{similar(k_i)}$$

This experiment is a measurement of a similar level on the page. However, because the one of this by the person's judgment was large, these evaluation values were used to measure the parameter change quantitatively.

3.3.2.2. Search condition

The object data of the output number is 1-50 to the retrieval key by ″data mining″, ” knowledge science” ,″.

Especially, when not describing it clearly, the number of selection words is assumed to be 100. A neighborhood distance is 4, and a study frequency is 500. The order of sorting of the output result is assumed to be a score. Table 9 and Table 10 are the retrieval results in the above-mentioned search condition.

1 Kenji Satou's Bookmarks (佐藤賢二のブックマーク)
2 Online Resources
3 Temporally
4 Genetic Knowledge Systems Laboratory Official Homepage
5 Jaist Knowledge Science
6 Jaist Knowledge Science
7 Activities
8 Bao Tu Ho
9 Publication
10 Bao Tu Ho
11 Research Area
12 Advanced Topics in Database Systems
13 データベース特論
14 Bao Tu Ho
15 Bao Tu Ho
16 i430.html
17 Takeshi Nagashima のブックマーク
18 Research Projects
19 PDCAT'2002 (Sep, 3-6, 2002, Kanazawa)
20 Hong Shen

21 Hong Shen
22 Tu Bao Ho
23 Susumu Horiguchi
24 Susumu Horiguchi
25 ks-master titles
26 ks-master titles
27 Bookmarks for
28 ks-master titles
29 ks-master titles
30 Modeling and Simulation
31 Workshop
32 Dung Trong [guye
33 Dung Trong [guye
34 Other links
35 Bao Tu Ho
36 Bao Tu Ho
37 PCDON2001workshop.html
38 Bao Tu Ho
39 Dung Trong [guye
40 Dung Trong [guye
41 Videos
42 Papers published at 1998 (In Japanese)
43 Papers published in 2001
44 JAIST Journal-Title
45 Videos
46 CSLSP-e Title
47 OJ-Title
48 International Symposiums
49 Hong Shen
50 Hong Shen

Table 9 Title list

Recall	**60.33**
Precision	**55.11**

Table 10 Recall and Precision

3.3.2.3. Parameter change

The following tables are ones that the neighborhood distance, the study frequency, and the number of words were changed. A blue character is an initial value. Changing recall is a change the element of the group component. Changing precision is change of variety in the word in the group.

Table 11 is the one that the neighborhood distance was changed from 1 to 5. Changing recall is few. Changing precision is extreme.

It is 1 and 3 that precision is high. There were a lot of similar documents when the neighborhood distance was 1 and 3. And, it was thought that the reason was that the use word increased.

Then, it is thought that the neighborhood distances is 3 is suitable.

	1	2	3	4
Recall	65.06	42.46	56.18	47.95
Precision	69.06	59.96	73.56	42.81

Table 11 Relation between neighborhood distance and adaptability.

Table 12 is the one that the study frequency was changed from 50 to 500. Changing recall is usual. Changing precision is few.

The resemblance of an actual document considerably had the change though changing precision was few. Data is overcrowded when the study frequency is low. Therefore, it is easy to do the group composition. However, the word of the group varies because the document is not similar. Therefore, precision is not so changed.

There should be a lot of study frequencies.

	50	100	150	200	250	300	350	400	450	500
Recall	69.27	62.1	69.88	70.6	60.58	73.92	60.88	58.83	62.04	47.95
Precision	46.39	48.21	32.42	48.32	36.38	44.9	48.96	40.86	43.06	42.81

Table 12 Relation between adaptability and study frequency

Table 13 is the one that number of words was changed 100 to 500. Changing recall is few. Changing precision is few.

Number of a similar documents at 400 are more than the time of 100. The reason for the number of words is to be used for the similar inspection.

There should be a lot of number of words.

	100	200	300	400
Precision	47.95	45.51	50.6	54.57
Recall	42.81	44.21	48.32	40.76

Table 13 Relation between number of words

3.3.2.4. Comparison of Euclid distances

Here, it compares it by the result of BLSOM and the ranking. The validity of the system is evaluated by doing so.

A common comparison item to both does not exist. In this thesis, a similar document was decided by using the score of the word. The score of the word is used for the comparison item. And, the difference of the score of the word between documents is shown as the following the Euclid distances.

$doc_k(word_i)$: Score of $word_i$ which exists in $document_k$

$$\frac{1}{M^2}\sum_{k=1}^{M}\sum_{j=1}^{M}\sum_{i=1}^{N}(doc_k(word_i)-doc_j(word_i))^2$$

The average of the distance of the group is requested by the result of BLSOM. It compares it for the distance of the average of the ranking.

The result is the following tables.

	data mining		kdd		perl		nano technology	
	Group	Rank	Group	Rank	Group	Rank	Group	Rank
Score	0.0600	0.0893	0.0488	0.0811	0.0669	0.1347	0.1039	0.1300
Date	0.0637	0.0961	0.0521	0.0849	0.1037	0.1233	0.0991	0.1330
Date disc	0.0327	0.0822	0.0875	0.0889	0.0457	0.0842	0.0714	0.1049

Table 14 Comparison of distances of rank and group

This retrieval words are " **data mining** " and " **kdd**" and “perl” and “nano technology” and . In addition, the standard of the ranking was requested in not only the score but also date the order.

The group result both is smaller than the ranking. This shows that BLSOM collects small documents of the distance. It is shown that the discovery of a similar document is in a word easier than the ranking.

Chapter4.Conclusion and problem

4.1.Conclusion

It is a success to be adjacent and to display a similar document. However, it can be said only that it is easy for that to discover a similar document. The meaning of the document group is necessary to look for a necessary document It is thought that the meaning of the document group and the feature of the document group are the same. Then, the word used for the feature of the document group at high frequency is used. Or, it is possible to display it for the correlation of the document group as the means.

4.2.Problem

It is possible to obtain it by increasing the study frequency to obtain a good result, and increasing the number of words. However, because both also influence the computational complexity, it is not possible to increase it too much. In a word, it is difficult to acquire the best parameter.

Because the meaning of the group is not understood, it does not become judgment assistance.

Whether a certain document is included in which group is not understood. It is possible to deal if it uses the color.

Acknowledgements

I am indebted to the participants in the studies for their gracious cooperation. Thanks also go to professor Ho Tu Bao, associate professor Masato Ishizaki, associate professor Takayuki Ito, associate Nguyen Trong Dung, and other participants for their support, assistance, and efforts.

References

[1] T.Kohonen,:
“Self-Organizing Map”, Springer (1997).

[2] Abe T., Kanaya S., Kinouchi M., Kudo Y., Mori H. and Matsuda H., C.D. Carpio, T.Ikemura, :
Gene classification method based on batch-learning SOM, Genome Informatics Series No.10, 314-315,1999

[3] http://www.namazu.org

[4]http://www.google.com

[5]http://www.yahoo.com

[6]http://www.goo.ne.jp

[7] http://www.looksmart.com/

[8] http://www.gnu.org/software/wget/wget.html

[9] http://kakasi.namazu.org/

[10] http://chasen.aist-nara.ac.jp/index.html.ja

[11] http://www01.tcp-ip.or.jp/~furukawa/pnamazu/

[12] Gerard Salton:
“Automatic text processing:the transform,analysis,and retrieval of information by computer”, Addison Wesley,1989.

Appendix

1 mkdocv.pl

```
#! /usr/local/bin/perl5 -w
#mkdocv.pl
#MaKeDocumentVector -> mkdocv
{
	$tmp = "./$tmp" if (my $tmp = $0) !~ /[¥/¥¥]/;
	while ($tmp =~ /^(.*[¥/¥¥])/){
		unshift(@INC, $1);
		last if !-l $tmp;
		$tmp = $1 . $tmp if ($tmp = readlink($tmp)) !~ /^¥//;
	}
}
require 'nmzidx.pl';
sub setvector
{
		my $nw	= $_[0];
		my $i = $_[1];
		my $word = $nw->getword($i);
		my $documentvector = $_[2] ;
		my %list ;
		$nw->read(¥$word,¥%list);
		for $id(sort keys %list)
		{
			$$documentvector{$id}{$i} = $list{$id};
		}
		$documentvector ;
}
sub writevector
{
		my $dir = "." ;
		my $documentvector = $_[0] ;
```

```
        my $nmz = new nmzidx($dir,"r");
        my $nmzw = new nmzidx($dir,'w');
        my $nvw    = $nmzw->open_vector ;
        my $nfw    = $nmzw->open_flist ;
        my $nf       = $nmz->open_flist ;
        for $docid(sort { $a <=> $b }keys %$documentvector)
        {
                my %veclist ; my %flist;
                my $sdocid = $docid ;
                my $sum = 0; my $str    = "";
                $nf->seek($sdocid,0); $nf->read(¥%flist) ;
                for $wordno(sort { $a <=> $b }keys %{$documentvector->{$docid}})
                {
                        $veclist{$wordno}=$documentvector->{$docid}->{$wordno}     ;
                        $sum += $veclist{$wordno};
                        $str = $str . "[$wordno,$veclist{$wordno}]" ;
                }
                $nvw->write($docid,¥%veclist);
                ${$flist{'field'}}{'score'} = $sum ;
                ${$flist{'field'}}{'vector'} = $str ;
                $nfw->seek($sdocid,0); $nfw->write(¥%flist);
        }
        $nmzw->replace_db;
        $nmzw->close; $nmz->close;
}
main
{
        my $dir ='.' ;
        my $nmz = new nmzidx($dir, 's');
        my $nw = $nmz->open_word;
        my $i ;
        my $documetnvector ;
        my $word;
```

```
        $i = 0 ;
        $word = $nw->getword($i);
        while($word ne ''){
                $documentvector = setvector($nw,$i,$documentvector);
                $i ++ ;
                $word = $nw->getword($i);
        }
        writevector($documentvector);
        $nmz->close ;
}
```

2　initblsom.pl

```
sub initblsom
{
   my $blist ;
   my %clist =() ;
   $blist->{'maxword'} = 300 ;
   $blist->{'file1'} = "data/ar" . $$ . ".txt" ;
   $blist->{'file2'} = "data/pos" . $$ . ".txt" ;
   $blist->{'file3'} = "data/vr" .    $$ . ".wrl" ;
   $blist->{'file4'} = "data/html" . $$ . ".html" ;
   $blist->{'logfile'} =    "data/log" . $$ . ".txt" ;

   if ($Keys){
      $blist->{'xmax'} = int($Max/2) ;
      $blist->{'ymax'} = int($Max/2)
      $blist->{'r'} = 4 ;
      $blist->{'loopmax'} = 500 ;
      $blist->{'chp'} = 1.0    ;
      $blist->{'cntmin'} = 50 ;
      $blist->{'cntmax'} = 300 ;
      foreach $key (@Keys){
         last if (($Whence + $Max) <= $lndx && $Max) ;
         my($offset, $next, $summary, $dt, $st, $dd, $grep);
         $keyno=$key,$keydb= " unless ($keyno, $keydb) = split(/¥#/, $key);
         ++$lndx;    $IntSize = $DbIntSize{$keydb};

         if($Whence < $lndx){
            unless (defined $DbSize{"RESULT$keydb"}){
               $DbSize{"RESULT$keydb"} = 0;
               my $dbpath = $keydb? $keydb: $DbPath;
               my @result = ("", ".normal"); my $result;
               unshift(@result, ".$Result") if length $Result;
               for $result (@result){
```

```
        last if &opentfile("RESULT$keydb","$dbpath.result$result");
        last if &opentfile("RESULT$keydb","$Headname.result$result");
    }
}
if($DbSize{"RESULT$keydb"} && $DbVer2{$keydb}){
    $blist->{$lndx}->{'Docid'} = $keyno ;
    $blist->{$lndx}->{'No'} = $lndx ;
    my $lscore = &conv_result($keyno,$keydb,$lndx,$Score{$key},"score");
    my $ltitle = &conv_result($keyno,$keydb,$lndx,$Score{$key},"title");
    my $luri = &conv_result($keyno,$keydb,$lndx,$Score{$key},"uri");
    my $bv = &conv_result($keyno,$keydb,$lndx,$Score{$key},"vector");
    $blist->{$lndx}->{'score'} = $lscore ;
    $blist->{$lndx}->{'Title'} = $ltitle ;
    $blist->{$lndx}->{'Url'} = $luri;
    my %vlist = ();
    while(length($bv)> 0){
        $bv =~ /¥[(¥d+),(¥d+)¥]/ ;
        my $lk = int($2) / int($blist->{$lndx}->{'score'}) ;
        $vlist{$1} = $lk ;
        $bv = $' ;
    }
    my @hkeys = sort{$vlist{$b} <=> $vlist{$a} ||
                        length($b) <=> length($a) ||
                        $b cmp $a} keys %vlist ;
    my $sum = 0 ; my $cnt = 0 ;
    for $idx(@hkeys){
        last if((($sum >$blist->{'chp'})&&($cnt > $blist->{'cntmin'})) ||
                ($cnt > $blist->{'cntmax'})) ;
        $blist->{$lndx}->{'wscore'}->{$idx} = $vlist{$idx};
        $cnt++ , $sum += $sc ;
        $clist{$idx} = 0 if($clist{$idx} < 1);
        $clist{$idx} ++ ;
    }
```

```
                $blist->{$lndx}->{'words'} = $cnt ;
                next;
            }
        }
    }
    $blist->{'maxno'} = $lndx ;
    my $oclist ; my $cnt = 0;
    my @hkeys = sort{$clist{$b} <=> $clist{$a} ||
                        length($b) <=> length($a) || $b cmp $a} keys %clist ;
    for $idx(@hkeys){
      last if($cnt >= $blist->{'maxword'});
      $cnt ++ ; $oclist->{$idx} = $idx;
    }
    $oclist->{'size'} = $cnt ;
    &callblsom($blist,$oclist);
  }elsif (!$Phone){
     if (&opentfile(*FH, "$DbPath.tips") || &openfiles(*FH, 'tips')){
       &message(<FH>);
       close(FH);
     }
  }
}
```

3　callblsom.pl

```
sub callblsom
{
    my $blist = shift ;
    my $clist = shift ;

    my $line ;my $n = $blist->{'maxno'} ;
    my $m = $clist->{'size'} ; delete $clist->{'size'};

    $blist->{'end'} = $n ;
    $blist->{'wordcount'} = $m ;
    if($n % $Max == 0){$blist->{'start'} = $n - $Max + 1 ;}
    else{$blist->{'start'} = int($n / $Max ) * $Max +1 ;}
    my $data = $blist->{'end'} - $blist->{'start'} +1 ;
    open(OUT,"> $blist->{'file1'}") ;
    print OUT "$data $m $blist->{'r'} $blist->{'xmax'} ";
    print OUT "$blist->{'ymax'} $blist->{'loopmax'}¥n";
    for(my $i = $Whence    +1 ; $i <= $n ; $i ++){
        for    $idx(sort{$a <=> $b} keys %$clist){
            if($blist->{$i}->{'wscore'}->{$idx} > 0 ){
               $blist->{$i}->{'wcount'} ++ ;
               $blist->{$i}->{'oword'}->{$idx} = $idx ;
            }
            else{ $blist->{$i}->{'wscore'}->{$idx} = 0    ;}
            printf OUT "%2.6f ",$blist->{$i}->{'wscore'}->{$idx} ;
        }
        print OUT "¥n" ;
    }
    close(OUT);
    system("./blsom $blist->{'file1'} $blist->{'file2'}");

    open(IN,"<$blist->{'file2'}"); $idx = $blist->{'start'};
    while(<IN>){
```

```
    $line =    $_ ; $line =~ /i¥s+=¥s+(¥d+) x¥s+=¥s+(¥d+) y¥s+=¥s+(¥d+)/;
    $i = ($1);
            $blist->{$idx}->{'x'} = int($2) ; $blist->{$idx}->{'y'} = int($3) ;
            #$line = $_ ;
            $line =~ /.*¥s*r =¥s+(¥d+.*¥d*) g =¥s+(¥d+.*¥d*) b = ¥s*(¥d+.*¥d*)/ ;
            $blist->{$idx}->{'r'} = $1 ;
            $blist->{$idx}->{'g'} = $2 ;
            $blist->{$idx}->{'b'} = $3 ;
            $idx ++ ;
}
close(IN);
$blist->{'max'}->{'x'} = $blist->{'xmax'} ;
$blist->{'max'}->{'y'} = $blist->{'ymax'} ;
require 'mkvrml.pl' ; mkvrml($blist,$blist->{'file3'});

system("cp parts.html $blist->{'file4'}");
open(OUT,">> $blist->{'file4'}");
print OUT "<dl><br>¥n";
for($i = $blist->{'start'} ;$i <= $blist->{'end'} ; $i ++){
      print OUT "$i <a target=¥"_top¥" href=¥"$blist->{$i}->{Url}¥">";
      print OUT "$blist->{$i}->{Title} </a><br>¥n";
}
print OUT "</dl></body></html>¥n" ;
close(OUT);

&output("<dl>¥n");
&output("<table cellpading=0 cellspacing=0>¥n<tr>¥n");
&output("<td>¥n")
&output("<embed src=¥"$blist->{'file3'}¥"");
&output("width=640 height=480>¥n");
&output("</td>¥n");
&output("<td>¥n");
&output("<iframe src=¥"$blist->{'file4'}¥"" );
```

```
        &output("width=320 height=480 scrolling=yes>¥n");
        for($i = $blist->{'start'} ;$i <= $blist->{'end'} ; $i ++){
                &output("$i <a href=¥"$blist->{$i}->{Url}¥">");
                &output("$blist->{$i}->{Title} </a><br>¥n");
        }
        &output("</iframe><br>¥n");
        &output("</td>¥n");
        &output("</tr>¥n<table>¥n");
        &output("</dl>¥n");

}
```

4　mkvrml.pl

```
sub mkvrml
{
        my $data = shift ;
        my $file    = shift ;

        my $PI       = atan2(1,1) * 4;
        my $hPI    = $PI / 2 ;
        my $rad    = 64.75 / 180 ;
        my $tanS = (sin($PI * $rad)) / (cos($PI * $rad));
        my $parts = "parts.wrl" ;
        my $view,$back,$line;
        my $strSize,$num ;

        my $i,$xavg,$yavg,$xmax,$ymax,$xmin,$ymin;
        system("/bin/cp $parts $file");
        open(OUT,">> $file");
        $back->{'r'} = 1 ; $back->{'g'} = 1 ; $back->{'b'} = 1 ;
        print OUT "Back{backC    $back->{'r'} $back->{'g'} $back->{'b'}}¥n";
        $num = $data->{'end'} - $data->{'start'} ;
        $strSize = 1.5 if($num < 51) ;
        $strSize = 3 if($num > 50) ;
        $xmax = $data->{'xmax'} ;$ymax = $data->{'ymax'} ;
        $xavg = $yavg = $xmin = $ymin = 0 ;
        for($i= $data->{'start'} ; $i <= $data->{'end'} ;$i++)
        {
                my %obj ;
                if($data->{$i}->{'x'} >= 0 && $data->{$i}->{'x'} ne ''){
                        $xavg += $data->{$i}->{'x'} ;
                        $yavg += $data->{$i}->{'y'} ;
                        my $px = $data->{$i}->{'x'} ;
                        my $py = $data->{$i}->{'y'} ;
                        my $red = $data->{$i}->{'r'} ;
```

```
                    my $green = $data->{$i}->{'g'} ;
                    my $blue = $data->{$i}->{'b'} ;
                    my $no = $data->{$i}->{'No'} ;
                    my $url = $data->{$i}->{'Url'} ;

                    $obj{'point'} =    "objT $px $py 0";
                    $obj{'color'} =    "objC    $red $ green $blue";
                    $obj{'No'}         = "objNo ¥"$no¥"" ;
                    $obj{'Url'}        = "objUrl ¥"$url¥"" ;
                    $obj{'strSize'} = "objSzs $strSize";
                    print OUT "Obj{ ";
                    for my $field(values %obj){print OUT "$field ";}
                    print OUT "} ¥n";
            }
}
my $count = $data->{'end'} - $data->{'start'};
my $xl = $xmax - $xmin +6; my $yl = $ymax - $ymin +6;
my $xs = $xmin -2 ; my $ys = $min - 2 ;
my $xh = (($xmax - $xmin) +    2 ) /2 ;
my $yh = (($ymax - $ymin) + 2 ) /2;
$view->{'x'} = $xmax / 2 ;    $view->{'y'} = $ymax / 2 ;
$view->{'z'} = abs(sqrt($view->{'x'} ** 2 + $view->{'y'} **2) * $tanS) ;
$line->{'x'}->{'length'} = $xl ; $line->{'y'}->{'length'} = $yl ;
$line->{'x'}->{'point'} = "$xh $ys 0" ;
$line->{'y'}->{'point'} = "-2 $yh    0"    ;
$line->{'x'}->{'rot'} = "0 0 1 $hPI" ;
$line->{'y'}->{'rot'} = "0 0 1 0" ;
print OUT "View{viewT    $view->{'x'} $view->{'y'} $view->{'z'}}¥n";
print OUT "Line{lineT $line->{'x'}->{'point'} " ;
print OUT "lineL $line->{'x'}->{'length'} " ;
print OUT "lineR $line->{'x'}->{'rot'} }¥n" ;
print OUT "Line{lineT $line->{'y'}->{'point'} ";
print OUT "lineL $line->{'y'}->{'length'} " ;
```

```
        print OUT "lineR $line->{'y'}->{'rot'} }¥n" ;
        close(OUT);
}
```

5 matutil.h

```
#ifndef __MATUTIL_H
#define __MATUTIL_H
#ifndef SCALAR
#define SCALAR float
#endif   /*SCALAR */
typedef SCALAR *vector,**matrix;
vector   new_vector(int);
matrix   new_matrix(int,int);
void      free_vector(vector);
void      free_matrix(matrix);
double   innerproduct(int,vector,vector);
void      vecprint(vector,int,int,char*);
void      matprint(matrix,int,int,char*);
#endif   /* __MATUTIL_H */
```

6 matutil.c

```
/*************************************************************
            matutil.c
*************************************************************/
#include <stdio.h>
#include <stdlib.h>

#ifndef SCALAR
#define SCALAR float
#endif   /*SCALAR */

typedef SCALAR *vector,**matrix,***map;
void error(char *message)
{
            fprintf(stderr, "¥n%s¥n", message);   exit(EXIT_FAILURE);
}
vector newvec(int n)
{
```

```
        return malloc(sizeof(SCALAR) * n);
}
matrix newmat(int nrow, int ncol)
{
        int i;
        matrix a;
        a = malloc((nrow + 1) * sizeof(void *));
        if (a == NULL) return NULL;
        for (i = 0; i < nrow; i++) {
                a[i] = malloc(sizeof(SCALAR) * ncol);
                if (a[i] == NULL) {
                        while (--i >= 0) free(a[i]);
                        free(a);    return NULL;
                }
        }
        a[nrow] = NULL;
        return a;
}
vector new_vector(int n)
{
        vector v;
        v = newvec(n);
        if (v == NULL) error("Out of memory.");
        return v;
}
matrix new_matrix(int nrow, int ncol)
{
        matrix a;
        a = newmat(nrow, ncol);
        if (a == NULL) error("Out of memory.");
        return a;
}
void free_vector(vector v)
```

```
{
        free(v);
}
void free_matrix(matrix a)
{
        matrix b;

        b = a;
        while (*b != NULL) free(*b++);
        free(a);
}
double innerproduct(int n, vector u, vector v)
{
        int i, n5;
        double s;

        s = 0;    n5 = n % 5;
        for (i = 0; i < n5; i++) s += u[i]*v[i];
        for (i = n5; i < n; i += 5)
                s += u[i]*v[i] + u[i+1]*v[i+1] + u[i+2]*v[i+2]
                                + u[i+3]*v[i+3] + u[i+4]*v[i+4];
        return s;
}
```

7 libpca.h

```
#ifndef __PRINCOLIB_H
#define __PRINCOLIB_H
#include "matutil.h"
double house(int , vector) ;
void tridiagonalize(int, matrix, vector , vector ) ;
int eigen(int, matrix, vector, vector);
void princolib(int, int, matrix, matrix, vector,vector);
#endif /* __PRINCOLIB_H end */
```

8 libpca.c

```
/*************************************************************
        libpca.c
*************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <limits.h>
#include <string.h>
#include <math.h>
#include "matutil.h"
#define EPS             1E-6

#define MAX_ITER        100
double house(int n, vector x)
{
        int i;
        double s, t;
        s = sqrt(innerproduct(n, x, x));
        if (s != 0) {
            if (x[0] < 0) s = -s;
            x[0] += s;    t = 1 / sqrt(x[0] * s);
            for (i = 0; i < n; i++) x[i] *= t;
        }
```

```
        return -s;
}
void tridiagonalize(int n, matrix a, vector d, vector e)
{
        int i, j, k;
        double s, t, p, q;
        vector v, w;
        for(k = 0; k < n - 2; k++) {
            v = a[k];    d[k] = v[k]; e[k] = house(n-k-1, &v[k+1]);
            if(e[k] == 0) continue;
            for(i=k+1;i<n;i++) {
                s = 0;
                for(j=k+1;j<i;j++) s += a[j][i] * v[j];
                for(j=i;j<n;j++) s += a[i][j] * v[j];
                d[i] = s;
            }
            t = innerproduct(n-k-1, &v[k+1], &d[k+1]) / 2;
            for(i=n-1;i>k;i--) {
                 p = v[i]; q = d[i] - t * p; d[i] = q;
                for (j = i; j < n; j++)
                a[i][j] -= p * d[j] + q * v[j];
            }
        }
        if(n >= 2){
            d[n-2] = a[n-2][n-2]; e[n-2] = a[n-2][n-1];
        }
        if(n >= 1) d[n - 1] = a[n - 1][n - 1];
        for(k=n-1; k>=0;k--) {
            v = a[k];
            if(k < n - 2) {
                for(i = k + 1; i < n; i++) {
                    w = a[i]; t=innerproduct(n-k-1, &v[k+1], &w[k+1]);
                    for(j=k+1;j<n;j++)    w[j] -= t * v[j];
```

```
                }
            }
            for(i=0;i<n;i++) v[i] = 0;
            v[k] = 1;
        }

}

int eigen(int n, matrix a, vector d, vector e)
{
        int i, j, k, h, iter;
        double c, s, t, w, x, y;
        vector v;
        tridiagonalize(n, a, d, &e[1]);

        e[0] = 0;
        for(h=n-1;h>0;h--){
            j = h;
            while(fabs(e[j]) > EPS * (fabs(d[j-1]) + fabs(d[j]))){
                j--;
                if (j == h) continue;
                  iter = 0;
                  do{
                    if(++iter>MAX_ITER)return EXIT_FAILURE;
                      w=(d[h-1] - d[h]) / 2; t = e[h] * e[h];
                      s = sqrt(w * w + t);    if (w < 0) s = -s;
                    x = d[j] - d[h] + t / (w + s);    y = e[j + 1];
                    for(k=j;k<h; k++){
                        if(fabs(x) >= fabs(y)){
                            t = -y / x;c = 1 / sqrt(t * t + 1);s = t * c;
                        }
                       else{
                            t = -x / y;s = 1 / sqrt(t * t + 1);c = t * s;
```

```
                    }
                    w = d[k] - d[k+1];t = (w * s + 2 * c * e[k+1]) * s;
                    d[k] -= t;    d[k + 1] += t;
                    if(k > j) e[k] = c * e[k] - s * y;
                    e[k+1] += s * (c * w - 2 * s * e[k+1]);
                        for(i=0;i<n;i++){
                            x = a[k][i];y = a[k+1][i];
                         a[k][i] = c * x - s * y;a[k+1][i] = s * x + c * y;
                    }
                    if(k < h - 1){
                            x = e[k+1];    y = -s * e[k+2];e[k+2] *= c;
                    }
                }
            }while(fabs(e[h])>EPS * (fabs(d[h-1]) + fabs(d[h])));
        }

    for(k=0;k<n - 1;k++){
        h = k;    t = d[h];
        for(i=k+1;i<n;i++)
            if(d[i] > t) {    h = i;    t = d[h];    }
            d[h] = d[k]; d[k] = t;
              v = a[h];    a[h] = a[k];    a[k] = v;
    }
    return EXIT_SUCCESS;
}
void pca(int n,int m,matrix x,matrix q,vector lambda,vector work)
{

    int i, j, k, ndf,method;
    double s, t, percent;

    method = 2 ;
    ndf = n - (method != 0);
```

```
        for(j=0;j<m;j++)
        {
            t = 0;
            for(i=0;i<n;i++) t += x[j][i];
            t /= n;
            if(method != 0) for(i=0;i<n;i++) x[j][i] -= t;
            q[j][j] = innerproduct(n,x[j],x[j]) / ndf;
            s = sqrt(q[j][j]);
            if(method == 2){
                q[j][j] = 1;
                for(i=0;i<n;i++) x[j][i] /= s;
            }
        }

        for(j=0;j<m;j++)
        {
            for(k=0;k<j;k++){
                q[j][k] = q[k][j] = innerproduct(n,x[j],x[k]) / ndf;
            }
        }
        if(eigen(m,q,lambda,work)) error("Fail!!");

}
```

9　blsom.c

```
/*********************************************
*        Batch Learning Self Orgnazing Map     *
*                                              *
**********************************************/
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <limits.h>
#include <string.h>
#include <math.h>
#include <time.h>

#include "matutil.h"
#include "libpca.h"

typedef SCALAR  ***map;

#define READERROR -1.00E+37;
#define MISSING   -0.98E+37;
#define OUTMIN     1.00E-10;
#define MAX        1.00;
#define MIN       -1.00;
#define readerror(x) ((x) < -0.99E+37)
#define missing(x)   ((x) < -0.97E+37)

double getnum(FILE *datafile)
{
         double x;
         char *rest, s[83];

         do {
                  if (fscanf(datafile, "%81s%*[^ ¥t¥n]", s) != 1)
```

```
                        return READERROR;
        } while (strchr("0123456789+-.", s[0]) == NULL);
        if (s[0] == '.' && s[1] == '¥0') return MISSING;
        s[81] = '?';  s[82] = '¥0';  x = strtod(s, &rest);
        if (errno == 0 && *rest == '¥0' && fabs(x) <= 0.97E+37)
                return x;
        errno = 0;  return READERROR;
}

FILE *open_data(char *filename, int *addr_n, int *addr_m,
                int *adr_nr,int *adr_dx,int *adr_dy,int *adr_cnt)
{
        FILE *datafile;
        double x, y,nr,cnt,dx,dy;

        *addr_n = *addr_m = 0;
        if ((datafile = fopen(filename, "r")) == NULL) {
                fprintf(stderr, "Can't open data file ¥n");
                return NULL;
        }
        x = getnum(datafile);  y = getnum(datafile);
        nr = getnum(datafile); dx = getnum(datafile);
        dy = getnum(datafile);cnt = getnum(datafile);
        if (x <= 0 || x > INT_MAX || y <= 0 || y > INT_MAX) {
                fprintf(stderr, "Can't read data.¥n");
                fclose(datafile);  return NULL;
        }
        *addr_n = (int)x;  *addr_m = (int)y;
        *adr_nr = (int)nr ; *adr_dx = (int)dx;
        *adr_dy = (int)dy;*adr_cnt = (int)cnt;
        return datafile;
}
```

```
int read_data(FILE *datafile, int n, int m, matrix x)
{
        int i, j, err;
        unsigned long missings;
        double t;

        err = 0;  missings = 0;
        for (i = 0; i < n; i++) for (j = 0; j < m; j++) {
                if (err) {  x[j][i] = READERROR;  continue;  }
                t = getnum(datafile);  x[j][i] = (SCALAR)t;
                if (! missing(t)) continue;
                if (readerror(t)) {
                        fprintf(stderr, "Read error (%d,%d)¥n", i+1, j+1);
                        err = 2;
                } else missings++;
        }
        return err | (missings != 0);
}

map newmap(int nrow, int ncol ,int ndepth)
{
        int i,j,k;
        map a;
        vector b ;

        a = malloc((nrow + 1) * sizeof(void *));
        if (a == NULL) return NULL;
        for (i = 0; i < nrow; i++) {
          a[i] = malloc(sizeof(SCALAR) * ncol);
          if (a[i] == NULL) {
            while (--i >= 0) free(a[i]);
            free(a);  return NULL;
          }
```

```
            else{
              for(j=0;j<ncol;j++){
                b = new_vector(ndepth);
                a[i][j] = b;
                if (a[i][j] == NULL) {
                   while (--j >= 0) free(a[i][j]);
                   for(k=i-1;k>=0;k--)for(j=ncol -1;j>=0;j--)free(a[i][j]);
                   while(--i >= 0) free(a[i]);
                   free(a);  return NULL;
                }
              }
            }
          }
          a[nrow] = NULL;
          return a;
}
map new_map(int nrow, int ncol, int ndepth)
{
          map a;

          a = newmap(nrow, ncol,ndepth);
          if (a == NULL) error("Out of memory.");
          return a;
}
void getColor(int n,vector red,vector green,vector blue,vector max,vector min)
{
          int i,j ;
          int color ;
          float r,g,b;

          color = 256 ;
          for(i=0;i<n;i++){
            if(max[2] != min[2] && max[2] != -min[2]){
```

```
            r = (red[i] - min[2])/((max[2] - min[2])/color);}
          else{r = red[i] ;}
          if(max[3] != min[3] && max[3] != -min[3]){
            g = (green[i] - min[3])/((max[3] - min[3])/color);}
          else{g = green[i] ;}
          if(max[4] != min[4] && max[4] != -min[4]){
            b = (blue[i] - min[4])/((max[4] - min[4])/color);}
          else{b = blue[i] ;}
          red[i] = r /color ; green[i] = g /color ; blue[i] = b /color ;
        }
}
void init(int n,int m,matrix x,matrix q,matrix axis,vector max,vector min,
          vector sep,vector avgx,vector sigma,vector crr,vector crg,vector crb)
{
        int i,j,k ;
        int dx,dy,endq;
        float fnum0,fnum1 ;
        vector fdata,avq;
        matrix msig;

        sigma[0]=0;sigma[1] = 0;
        avq = new_vector(m);
        msig = new_matrix(2,m);
        for(i = 0; i < 5 ; i++){min[i] = MAX;max[i] = MIN;}
        for(i=0;i<n;i++){
          for (k = 0; k < m && k < 5; k++){
            axis[k][i] = 0;
            for (j = 0; j < m; j++){
              avgx[j] = 0;
              if(fabs(q[k][j]) > 1.00E-10){
                axis[k][i] += q[k][j] * x[j][i];
              }else{
             axis[k][i] += 0*x[j][i];
```

```
      }
    }
    switch(k){
    case 2 :crr[i] = axis[k-2][i];break;
    case 3 :crg[i] = axis[k-2][i];break;
    case 4 :crb[i] = axis[k-2][i];break;
    default :msig[k][i] += axis[k][i];
    }
    if(axis[k][i] < min[k]) min[k] = axis[k][i] ;
    if(axis[k][i] > max[k]) max[k] = axis[k][i] ;
  }
}
for(i=0;i<m;i++){
  avq[i] = 0;
  for(j=0;j<m;j++){
    if(q[j][i] == 1) break ;
    if(fabs(q[j][i]) > 1.00E-10) avq[i] += q[j][i];
    if(fabs(q[j][i]) <= 1.00E-10) avq[i] = avq[i];
  }
  if(i==j) break ;
}
endq = i ;
sigma[0] = 0 ;sigma[1] = 0 ;
for(i=0;i<n;i++){
  for(j=0;j<m;j++){
    avgx[j] +=  x[j][i];
    if(i == n-1){
      avgx[j] = avgx[j] / n ;
      sigma[0] += (q[0][j] - avq[j]/m)*(q[0][j] - avq[i]/m);
      sigma[1] += (q[1][j] - avq[j]/m)*(q[1][j] - avq[i]/m);
    }
  }
}
```

```
        sigma[0] = sigma[0] / (m-1) ; sigma[1] = sigma[1] /(m-1);
        dy = (int)(sep[0]*sigma[1]/sigma[0] +1);
        if(dy <(int) sep[0] * 2 / 3 ) dy = (int) sep[0] * 2 /3 ;
        sep[1] = dy ;dx=(int)sep[0];
        free_matrix(msig);
        free_vector(avq);
}
void initdata(int n,int m,int dx,int dy,vector sigma,
             matrix q,map field ,vector avgx)
{
        int i,j,k;
        float fnum0,fnum1;
        vector fdata ;
        for(i=0;i<dy;i++){
          for(j=0;j<dx;j++){
            fdata = new_vector(m);
            for(k=0;k<m;k++){
              if(fabs(q[0][k]) > 1.00E-10){
                fnum0 = q[0][k] * (float)((float)(j - (float)(dx / 2)) / dx) ;
              }else{
                fnum0 = 0 ;
              }
              if(fabs(q[1][k]) > 1.00E-10){
                fnum1 = q[1][k] * (float)((float)(i - (float)(dy / 2)) / dy) ;
              }else{
                fnum1 = 0 ;
              }
              fdata[k] = avgx[k] + 5*sigma[0]*(fnum0 + fnum1) ;
            }
            field[i][j] = fdata ;
          }
        }
}
```

```
void search(int datano,int endx,int endy,int m,vector data,
             matrix lattice,map field,int *x,int *y,float *min)
{
          int i,j,k;
          vector check;
          float sum,small ;
          for(i=0;i<endy;i++){
            for(j=0;j<endx;j++){
              check = field[i][j];
              small = *min;
              sum = 0 ;
              for(k=0;k<m;k++){
                sum += (check[k] - data[k])*(check[k] - data[k]) ;
                if(sum > small * small) break;
              }
              sum = sqrt(sum);
              if(small > sum){
                *x = j ; *y = i; *min = sum ; lattice[i][j] = datano ;
              }
            }
          }
}
void comdata(int m,vector data,vector add1,int nm1,
             vector add2,int nm2,vector add3,int nm3,vector add4,int nm4){
          int i;

          for(i=0;i<m;i++){
            if(add1 != NULL) add1[i] = (add1[i]*nm1 + data[i]) / (nm1 + 1);
            if(add2 != NULL) add2[i] = (add2[i]*nm2 + data[i]) / (nm2 + 1);
            if(add3 != NULL) add3[i] = (add3[i]*nm3 + data[i]) / (nm3 + 1);
            if(add4 != NULL) add4[i] = (add4[i]*nm4 + data[i]) / (nm4 + 1);
          }
}
```

```
void adddate(int ex,int ey,int x,int y,int m,int r,
             vector data,matrix lattice,map avw)
{
          int i,j,k,ij;
          int nm1,nm2,nm3,nm4;
          vector add,add1,add2,add3,add4;

          add1 = avw[y][x];
          nm1 = lattice[y][x];
          add2 = NULL,nm2 = 0; add3 = NULL,nm3 = 0 ; add4 = NULL,nm4 = 0;
          comdata(m,data,add1,nm1,add2,nm2,add3,nm3,add4,nm4);
          lattice[y][x] ++ ;

          for(i=1;i<=r;i++){
            for(j=i;j>=1;j--){
              ij = i - j;
              add1 = NULL,add2 = NULL ; add3 = NULL ; add4 = NULL;
              nm1 = 0,nm2 = 0, nm3 = 0 ,nm4 = 0;
              if(x+ij < ex && y+j  < ey ){
                nm1 = lattice[y+j][x+ij] ;
                add1 = avw[y+j][x+ij];
                lattice[y+j][x+ij] ++;
              }
              if(x+j  < ex && y-ij >= 0 ){
                nm2 = lattice[y-ij][x+j] ;
                add2 = avw[y-ij][x+j];
                lattice[y-ij][x+j]++ ;
              }
              if(x-ij >= 0 && y-j  >= 0 ){
                nm3 = lattice[y-j][x-ij] ;
                add3 = avw[y-j][x-ij];
                lattice[y-j][x-ij]++ ;
              }
```

```
            if(x-j  >= 0 && y+ij < ey ){
              nm4 = lattice[y+ij][x-j] ;
              add2 = avw[y+ij][x-j];
              lattice[y+ij][x-j]++ ;
            }
            comdata(m,data,add1,nm1,add2,nm2,add3,nm3,add4,nm4);
          }
        }
}
void update(int dx,int dy,int m,int radius,
            float w,matrix lattice,map field,map avw)
{
        int i,j,k,hx,hy;
        vector dt1,dt2,dt3,dt4,up1,up2,up3,up4;

        hx = dx /2 ; hy = dy /2 ;

        for(i=0;i<hy;i++){
          for(j=0;j<hx;j++){
            dt1 = avw[i][j]      ; up1 = field[i][j]      ;
            dt2 = avw[i][j+hx]   ; up2 = field[i][j+hx]   ;
            dt3 = avw[i+hy][j]   ; up3 = field[i+hy][j]   ;
            dt4 = avw[i+hy][j+hx]; up4 = field[i+hy][j+hx];
            for(k=0;k<m;k++){
              if(lattice[i][j] != 0)       up1[k] += w*(dt1[k] - up1[k]);
              if(lattice[i][j+hx] != 0)    up2[k] += w*(dt2[k] - up2[k]);
              if(lattice[i+hy][j] != 0)    up3[k] += w*(dt3[k] - up3[k]);
              if(lattice[i+hy][j+hx] != 0) up4[k] += w*(dt4[k] - up4[k]);
            }
            field[i][j]    = up1   ;field[i][j+hx]    = up2 ;
            field[i+hy][j] = up3   ;field[i+hy][j+hx] = up4 ;
          }
          if(2*hx != dx){
```

```
            dt1 = avw[i][dx-1] ; up1 = field[i][dx -1];
            for(k=0;k<m;k++){
              if(lattice[i][dx-1] != 0) up1[k] += w*(dt1[k] - up1[k]);
            }
            field[i][dx-1] = up1 ;
          }
        }

        if(2*hy != dy){
          for(j=0;j<hx;j++){
            dt1 = avw[dy -1][j]      ; up1 = field[dy -1][j]      ;
            dt2 = avw[dy -1][j+hx]   ; up2 = field[dy -1][j+hx]   ;
            for(k=0;k<m;k++){
              if(lattice[dy-1][j] != 0)       up1[k] += w*(dt1[k] - up1[k]);
              if(lattice[dy-1][j+hx] != 0)    up2[k] += w*(dt2[k] - up2[k]);
            }
            field[dy-1][j]    = up1   ;field[dy-1][j+hx]    = up2 ;
          }
          if(2*hx != dx){
            dt1 = avw[dy-1][dx-1] ; up1 = field[dy-1][dx -1];
            for(k=0;k<m;k++){
              if(lattice[dy-1][dx-1] != 0) up1[k] += w*(dt1[k] - up1[k]);
            }
            field[dy-1][dx-1] = up1 ;
          }
        }
}
float getweight(int t,int T,float w0){
        float cw,vw;
        cw = 0.01;
        vw = 0.06*(1-t/T);
        return  cw > vw ? cw : vw ;
}
```

```
int main(int argc,char *argv[])
{
        int n,m,method,i,j,k,radius,count,dx,dy;
        int loop,px,py ;
        float weight,w0,lmin;
        float sx,sy,bx,by,s;
        vector lamda,work ;
        vector sep,near;
        vector max,min;
        vector crr,crg,crb ;
        vector sigma,avgx;
        matrix axis,inx,x,q;
        matrix lattice,ulattice;
        map field,avw;
        FILE *datafile ;
        time_t seed ;

        datafile = open_data(argv[1],&n,&m,&radius,&dx,&dy,&count);
        if(datafile==NULL)error("¥nData error ¥n");
        x = new_matrix(m,n);q = new_matrix(m,m) ;inx = new_matrix(n,m);
        if(read_data(datafile,n,m,x))error("¥nData error ¥n");
        fclose(datafile);

        lamda = new_vector(m) ; work = new_vector(m);
        axis = new_matrix(5,n);
        sep = new_vector(2); sep[0]=dx;/*sep[1]=dy; */
        avgx = new_vector(m);
        crr = new_vector(n);crg = new_vector(n);crb = new_vector(n);
        max = new_vector(5);min = new_vector(n);
        sigma = new_vector(2);

        pca(n,m,x,q,lamda,work) ;
```

```
init(n,m,x,q,axis,max,min,sep,avgx,sigma,crr,crg,crb);
dx =(int)sep[0];dy=(int)sep[1];
field = new_map(dy,dx,m);avw = new_map(dy,dx,m);
initdata(n,m,dx,dy,sigma,q,field,avgx);
getColor(n,crr,crg,crb,max,min);
for(i=0;i<m;i++)for(j=0;j<n;j++)inx[j][i] = x[i][j] ;

if((max[0] == -min[0])&&(max[1] == -min[1])&&(max[2] == -min[2])){
  for(i=0;i<n;i++){
    axis[2][i]=(int)x[0][i];
    axis[3][i]=(int)x[1][i];
    time(&seed);srand(seed % UINT_MAX);
    crr[i]=rand();crb[i]=rand();crg[i]=rand();
  }
}else{
  loop = 1 ;
  lattice = new_matrix(dy,dx);ulattice = new_matrix(dy,dx);
  while(loop <= count ){
    for(i=0;i<sep[1];i++)for(j=0;j<sep[0];j++)lattice[i][j] = 0;
    for(i=0;i<n;i++){
      px = 0 ; py = 0 ; lmin = 9999.0 ;
      search(i,dx,dy,m,inx[i],lattice,field,&px,&py,&lmin);
      adddate(dx,dy,px,py,m,radius,inx[i],lattice,avw);
      axis[2][i] = px ; axis[3][i] = py ;
    }
    weight = getweight(loop,count,w0);
    update(dx,dy,m,radius,weight,lattice,field,avw);
    loop ++;
  }
}

datafile = fopen(argv[2],"w");
for(i=0;i<n;i++){
```

```
            px = (int)axis[2][i] ; py = (int)axis[3][i];
            fprintf(datafile,"i = %5d x = %8d y = %8d ",i,px,py);
            fprintf(datafile,"r = %2.5f ",crr[i]);
            fprintf(datafile,"g = %2.5f ",crg[i]);
            fprintf(datafile,"b = %2.5f ¥n",crb[i]);
        }
        fclose(datafile);
}
```