A Method Of Data Queries Based On Data Visualization Techniques

Hirotaka Mizuno, Yoji Taniguchi, Hiroshi Tsuji, Hiroshi Yajima

Systems Development Laboratory
Hitachi, Ltd.
8-3-45 Nankouhigashi, Suminoe-Ku
Osaka 559
JAPAN

Norihisa Komoda

Department of Information Systems Engineering
Faculty of Engineering, Osaka University
2-1 Yamadaoka, Suita
Osaka 565
JAPAN

Abstract: This paper proposes a "visual query interface," which supports users to search and manipulate data in databases without knowing about their data structure. As basic functions of the visual query interface, data visualization, image zooming, and dynamic filtering are proposed. The data visualization helps users understand data more easily from various viewpoints. The image zooming allows users to understand visualized data in various resolutions. And the dynamic filtering supports users' trial-and-error in searching data, and its user interface is integrated with the visualization.

Keywords: Data queries, data visualization, user interface, database, filtering

Hirotaka Mizuno received his BE and ME degrees from Osaka University in 1979 and 1981, respectively. He is currently Senior Research Engineer at Systems Development Laboratory, Hitachi, Ltd., Osaka. His research interests include pattern processing and decision support systems. He is a member of the Information Processing Society of Japan, and of the Institute of Electrical Engineers of Japan.

Yoji Taniguchi received his BE and ME degrees from Waseda University in 1987 and 1989 respectively. He is currently a research engineer at Systems Development Laboratory, Hitachi, Ltd., Osaka. His research interests include decision support systems, data visualization, and data mining. He is a member of the Information Processing Society of Japan, and of the Institute of Electrical Engineers of Japan.

Hiroshi Tsuji received his BE, ME, and Ph.D degrees from Kyoto University in 1976, 1978 and 1993, respectively. He is currently Senior Research Engineer at Systems Development Laboratory, Hitachi, Ltd., Osaka. His research interests include knowledge-based systems, groupware, information retrieval systems, and human-computer interaction. He is a member of the ACM, the IEEE, and of the Japanese Society of Artificial Intelligence.

Hiroshi Yajima received his BE, ME, and Ph.D degrees from Kyoto University in 1973, 1975 and 1992, respectively. He is currently Department Manager of the Kansai Systems Laboratory at Systems Development Laboratory, Hitachi, Ltd., Osaka. His research interests include information systems planning and decision support systems. He is a member of the IEEE, the Information Processing Society of Japan, and of the Institute of Electrical Engineers of Japan.

Norihisa Komoda received his BE, ME, and Ph.D degrees from Osaka University in 1972, 1974 and 1982, respectively. He is currently Professor at the Department of Information Systems Engineering, Faculty of Engineering, Osaka University. He stayed at the University of California, Los Angeles, as a visiting researcher from 1981 to 1982. His research interests include systems engineering and knowledge information processing. He is a member of the IEEE, the ACM, the Institute of Electrical Engineers of Japan, and of the Society of Instrument and Control Engineering. He received the Awards for outstanding paper and the Awards for outstanding technology both from the Society of Instrument and Control Engineering.

1. Introduction

As small and powerful personal computers spread to offices, it has been common that end-users operate computers to access databases (DBs) by themselves. For extracting data from databases, users must use an interactive query interface. For the query process to proceed smoothly, functionality and usability of the query interface are crucial. Currently, relational database (RDB) is a mainstream in databases. As query interfaces for RDB, SQL (Structured Query Language) [1] and QBE (Query By Example) [2, 3] are commonly used. Since data are stored as tables in RDB, the above query interfaces are designed to manipulate table-form data. As a result, users have to know that the data to be manipulated are in table form, which makes it difficult to use. This paper proposes a "visual query interface," which supports users to search and manipulate data without knowing about their data structure. Two facilities of the interface are presented: data visualization that allows users to understand data more easily, and filtering that searches certain data in an environment integrated with the visualized images.

2.Conventional Query Interfaces

Figure 1 shows the procedure how users search data from databases. This procedure can be regarded as a task hierarchy introduced by Norman [4]. In execution, the following stages should be completed:

1. Forming the goal at business term level (e.g. examining names of male employees)
2. Forming the intention to act at DB term level (e.g. retrieving records whose sex value is male from table "Employee")
3. Specifying the sequence of actions (e.g. specifying SQL commands)
4. Executing the action sequence (e.g. operating computer with keyboard or mouse)

In evaluation, the following stages should be completed:

5. Perceiving the state of the world (e.g. confirming that the commands terminated successfully)
6. Interpreting the state of the world (e.g. confirming that the Table is displayed as a result)
7. Evaluating the outcome (e.g. confirming that sex values of all the records are male)

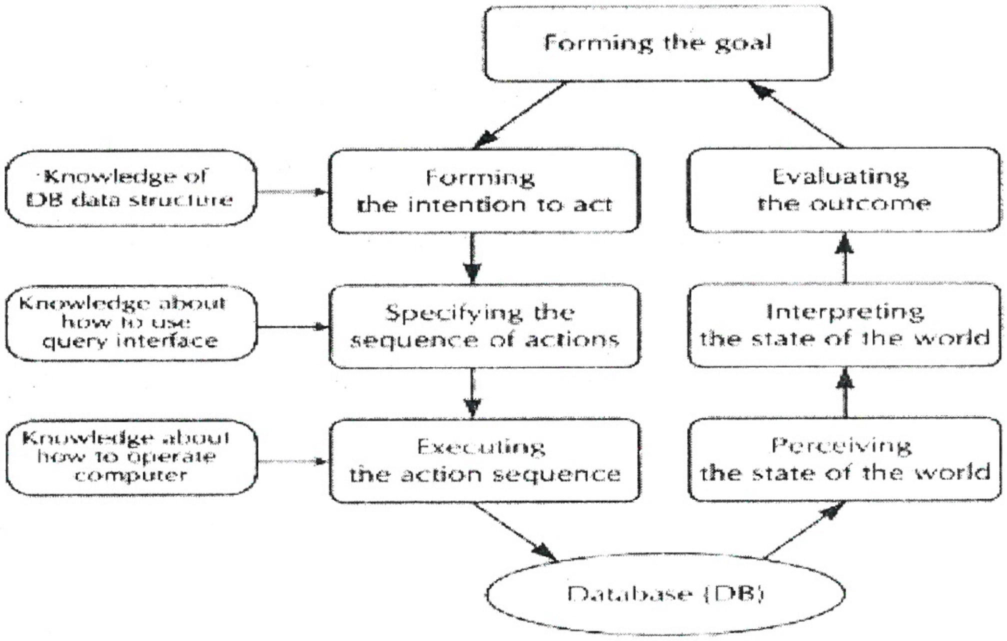

Figure 1. Stages of Action and Evaluation in Data Query

8. Evaluating whether the goal is achieved (e.g. examining the name values in the retrieved records)

As shown in the Figure above, the following types of knowledge are necessary in the above stages:

(a) knowledge of DB data structure:

(at stage 2 and stage 7)

Usually, data structures in DB do not coincide with those users have beforehand imagined . Consequently, users are required to learn the data structure in DB and understand the difference between these two structures. As for RDB, users have to know field names of the Tables stored.

(b) knowledge about how to use query interface: (at stage 3 and stage 6)

This is about commands or syntax of a specific query interface.

(c) knowledge about how to operate computer: (at stage 4 and stage 5)

This is about how to operate a specific computer and the meanings of the messages from it.

To remember that the knowledge (a) and (b) is a big burden to users.
In SQL, for example, commands are described in the following manner:

```
select Name from Employee
                where Age >= 40
```

In the sense that SQL's commands look natural to users, it succeeds in decreasing the users' burden to some extent. However, users still have to remember field names of data stored in DB. Moreover, users have to transform the data structures in between DB and their mind. In QBE, by specifying the Table name, users can see the Table structure (i.e. table name and field names) on screen as shown in Figure 2 (a). This absolves users of the obligation of remembering data structures. By writing simple commands into cells, users can specify search conditions. In this manner, QBE overcomes many of the problems that SQL has. However, the structure of data is still tightly fixed in table form.

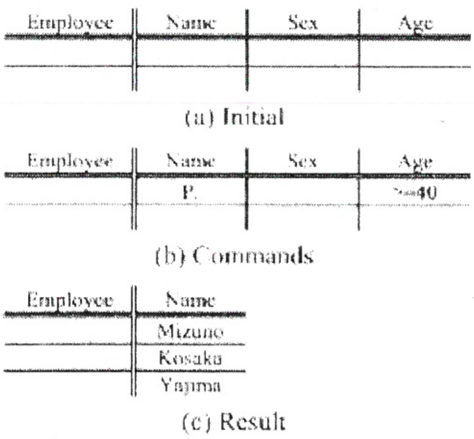

Figure 2. Example of QBE

3. Basic Idea of Visual Query Interface

As described in the previous Section, problems to overcome in data queries are summarized as follows:

(1) How easy to understand data?

Displays of original data and results are fixed in table forms that are the data structures in RDB. Users are requested to have mapping between them and the structures they imagine in their mind.

(2) How- easy- to- specify commands to manipulate or retrieve data?

Operations for data queries are irrespective of those displaying data, or both are fixed in table form. This makes it difficult for users to remember the usage.

As a query interface that overcomes these problems, a "visual query interface" is proposed, having the following features [5]:

(1) Providing data visualization facilities:

This transforms appearances of both the original data and the queried data into those whose structures are similar to what users would imagine. By hiding data structure in DB, users can understand the data easily. Moreover, this also enables users to search data by browsing without any commands.

(2) Providing user interface for searching data as an environment integrated with the visualized images:

This relates methods of manipulating and searching data to displays of visualized image, which narrows the gap between execution and evaluation in the data query process.

To realize the above, the following basic functions are proposed:

- Visualizing of data:
- for supporting users to view and understand data from various viewpoints in searching them.

- Image zooming of visualized data:
- for supporting users to view data in various resolutions in searching them.
- Filtering:
- for searching and extracting data for users.
- Sorting:
- for supporting users to understand the order of data according to certain fields of record.

By combining the functions flexibly, a variety of ways of searching data can be provided to users. For instance, by displaying the original data and changing only the appearances of the filter-passed data, users can easily know the data next to the other data that satisfy certain conditions.

Figure 3 shows an overview of the proposed visual query interface and its relation to a conventional query interface. In the visual query interface, a visualizing unit, a filtering and sorting unit for preprocessing, and an image zooming unit for postprocessing are connected in series. The visualizing unit generates images including the filtering results. The image zooming unit generates images in various resolutions. As shown in the Figure below, the object data to the visual query interface are what is retrieved under rough conditions with the conventional query interface. Even if the proposed user interface is flexible and friendly enough, it cannot be applied to processing one million of data. It is because that for users to search data on the visualized image of one million data is not practical. As a practical use, the proposed user interface can be applied to the preprocessed data that are primarily retrieved with the conventional query interface. The visual query interface is supposed to treat up from several hundreds to several thousands of data. In this respect, the interface is not replacing the conventional one but complementing it.

4. Data Visualization From Various Viewpoints

4.1 Classification of Data According to the Information They Represent

Most of the data in databases can be considered as representing information about objects existing in the real world. The visualization methods used in the visual query interface are proposed from this point of view. For supporting users to understand data, fields of record data in databases are naturally classified into categories listed in Table 1, according to the information they represent:

(a) Data field representing contents of objects:

Including invisible properties such as name, age, and so on.

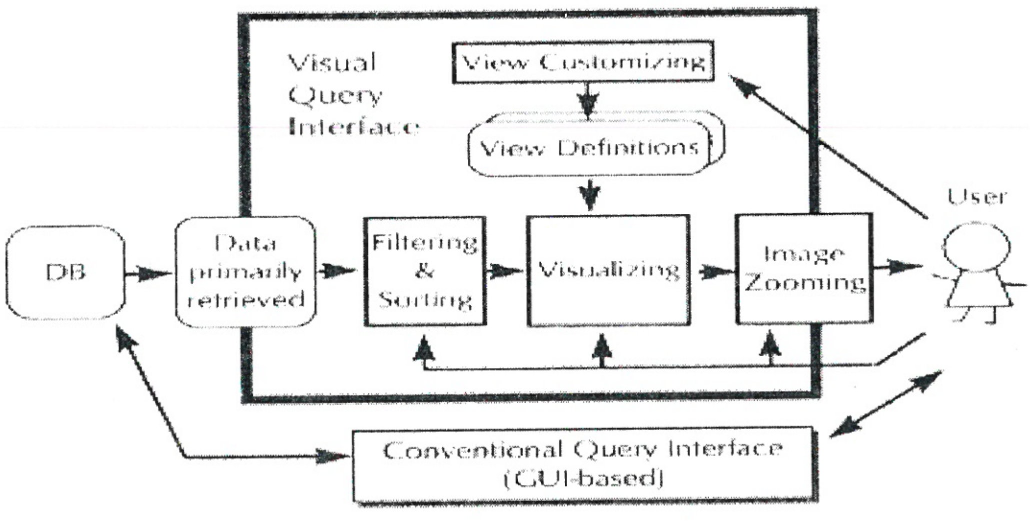

Figure 3. Visual Query Interface

(b) Data field representing appearances of objects:

Including visible properties such as height, portrait image, and so on.

(c) Data field representing locations of objects:

Including spatial or temporal properties such as position (coordinates), address, seat, date, and so on. Also including relative location like front-rear relation.

(d) Data field representing relations with other objects:

Including hierarchy, and so on. This can be understood as generalized location.

Table 1. Classification of Data According To Information Represented

category / object	contents	appearance	location	relation
person	name, gender, date of birth	height, portrait	address, desk	parent-child, company-section
book	contents, title	cover, size	rack, library	area, author
article	contents, title	headline	occurrence time	area, writer

4.2 Methods of Data Visualization for Each Category

Data visualization methods for each category are shown as follows:

(a) Data field representing contents:

Objects can be displayed as stereotypes that users imagine in their mind, such as

- table form for numerical values, or
- icons associated with differences of sex or age [6, 7, 8].

(b) Data field representing appearances:

Objects can be displayed as they are expected to look like, such as

- reflecting color properties to those of icons or name labels, or
- reflecting length or size properties to those of icons, or
- attaching their portrait properties to icons.

(c) Data field representing locations:

Objects can be arranged on layout image according to their location properties, like

- arranging icons or labels on map image, or
- arranging icons or labels on time scale [9, 10].

(d) Data field representing relations:

- arranging icons or labels with line segments connecting them (in case their relation is a hierarchy, the visualized image will make up a tree-structured map [9, 11]).

Two examples of visualizing data are shown in Figure 4 and Figure 5. In these examples, the object data represent employees in a company. The data are a set of records whose fields are name, sex, year of entrance, section, seat location, and so on. Figure 4 shows the visualized data as two types of icons that represent the difference of sex fields for helping users intuitively understand each employee's sex. On the other hand, Figure 5 shows the visualized data as name labels positioned in a room layout image according to their seat location fields, for helping users understand the employees' spatial relations about their seat locations.

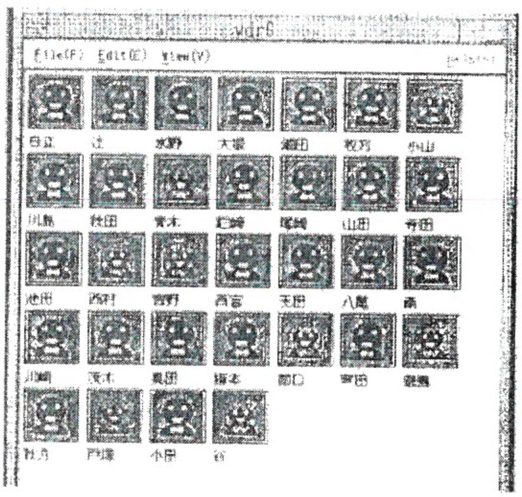

Figure 4. Visualized Employee Records As Icons Associated with Sex Field

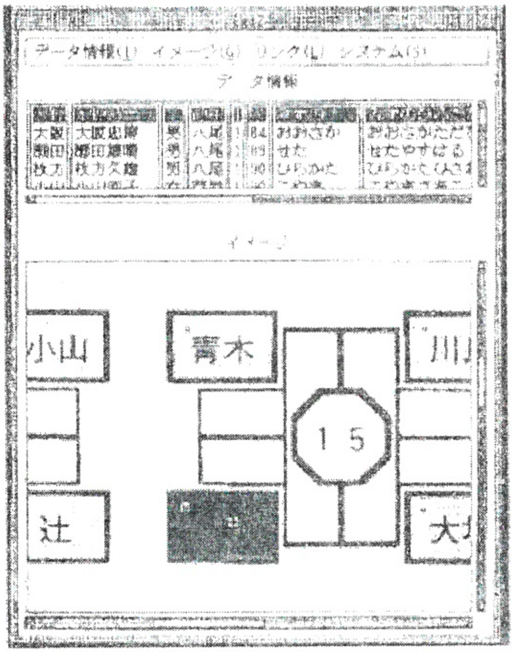

Figure 5. Visualized Employee Records As Name-labels on A Layout Image

5. Zooming of Visualized Image

The information that users can acquire from visualized data varies as the resolution of visualized image changes. Each is useful for different purposes. By zooming out the visualized images, many data are displayed as points, and users can understand their distribution over a wide area. On the contrary, by zooming in the visualized images, data over a partial area are displayed in detail, and users can understand the details there. Figure 6 shows the example of image zooming of the visualized data of Figure 5, which visualizes the two layout images in different resolutions. In Figure 6, one image displays all the data and the other displays the partial area in detail at the same time. On the entire image in the left window, a rectangle is shown which corresponds to the partial image in the right window, so that users may navigate smoothly without losing their way.

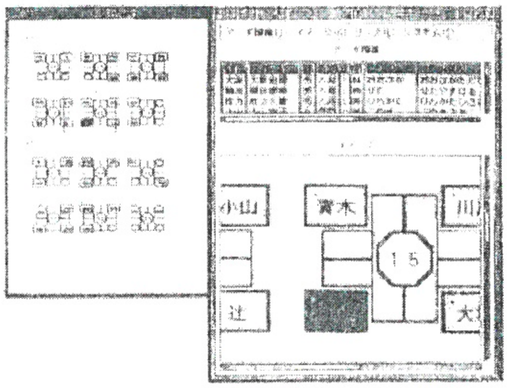

Figure 6. Display of Entire Outline and Partial Detail

6. Dynamic Filtering and Its User Interface

Filtering is the function that searches data which satisfy certain conditions. By adding postprocessing to the results of filtering, we propose a variety of usages for various purposes. The methods of postprocessing after filtering and their usage are as follows:

- Displaying only the data that passed the filter:

  to extract data satisfying certain conditions.

- Displaying the filter-passed data with their appearances changed on original data (e.g. high-lighted or color-changed):

  to help users understand which data satisfy certain conditions.

- Marking the filter-passed data:

  to help users remember them for preparation of future use.

- Extracting the marked data:

  to extract the memorized data for their reuse.

As a filtering facility in the visual query interface, "dynamic filtering" is proposed, which performs the above variations of filtering dynamically . Figure 7 shows the flowchart of dynamic filtering. Every time the slider or scale bar is dragged by users to change value of specific field in search conditions, filtering is invoked automatically. Filtering searches the records satisfying the updated conditions, and the results are reflected to the visualized image in real-time according to the prespecified instruction about visualization. This allows users to do "trial-and-error" rapidly in searching with ambiguous conditions like "I want to search about ten hotels whose rate is around 10,000 yen." In addition, users' load of operation is lightened.

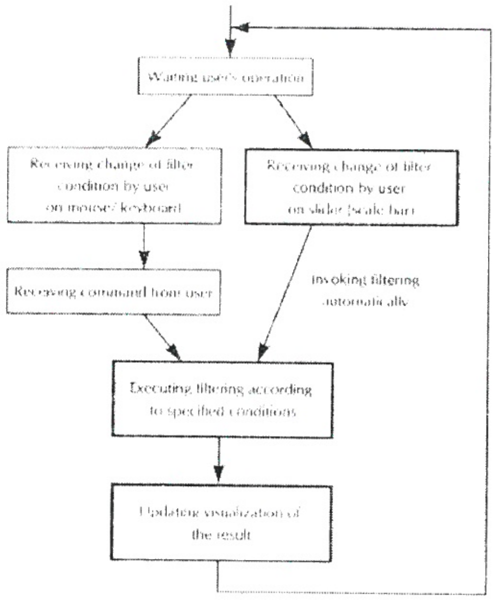

Figure 7. Flow of Dynamic Filtering

Figure 8 shows an example of dynamic filtering. In this example, the records of employees who are male and young are extracted from all the records. In this case, filtering is executed dynamically as users change the search conditions [12]. As the slider is dragged and the field value of entrance year is changed, filtering is automatically executed and the result is updated on screen. In the example, the icons representing records of male employees who entered after 1988 are displayed. The dynamic filtering is useful in case that users search data by repeating changing search conditions slightly in a "trial-and-error" manner.

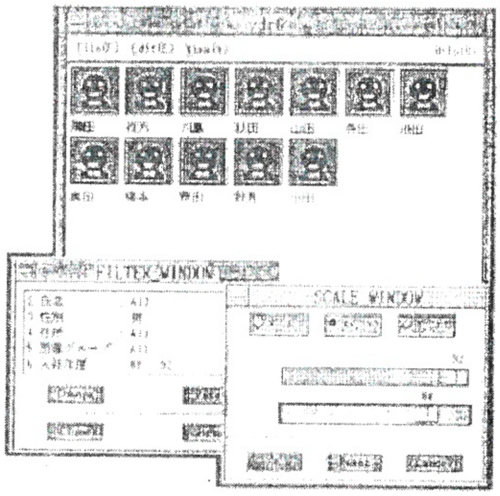

Figure 8. Result of Filtering Employee Records Using Dynamic Filtering

Next, a method of filtering records positioned on images and its user interface are presented. Figure 9 is an example of visualized image of person records on a map image. So far, another window has been used for users to specify the search conditions and invoke filtering. However, in this method, the visualized result of filtering has little relationship with user interface of filtering. To improve this, the idea of "filter-sheet" is presented.

Filter-sheet is an area with search conditions embedded. In Figure 9, the sheet is defined as a rectangle area. By users' placing it on the image, records are searched which satisfy the search conditions within the sheet, and results are displayed on the image. In the example, records of the persons whose age is older than 70 are displayed as result. As shown in the flowchart in Figure 10, every time the sheet is moved or re-sized by users' operation, the filtering is automatically invoked and the result is updated on image. With filter-sheet, users can search records in the environment integrated with the visualized images.

As a natural extension of this idea, plural filter-sheets can be used together. For example, by defining one sheet for searching persons whose age is older than 70 and another sheet for searching male together, records that satisfy both conditions are dynamically searched in the overlapping area. There is a natural mapping between the overlapping of sheets and the combination of conditions. In this sense, the proposed user interface for filtering is a user friendly one.

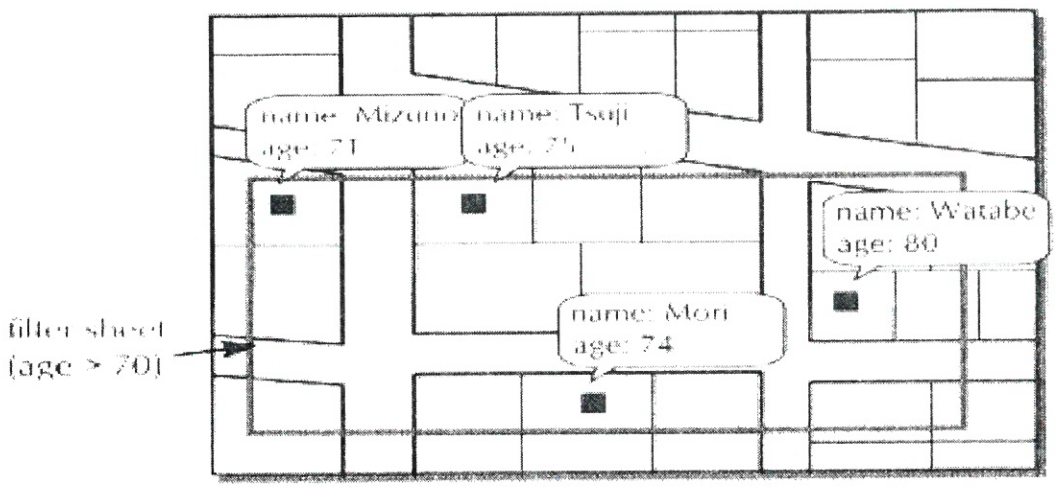

Figure 9. Result of Filtering Records on Map Image Using Dynamic Filtering

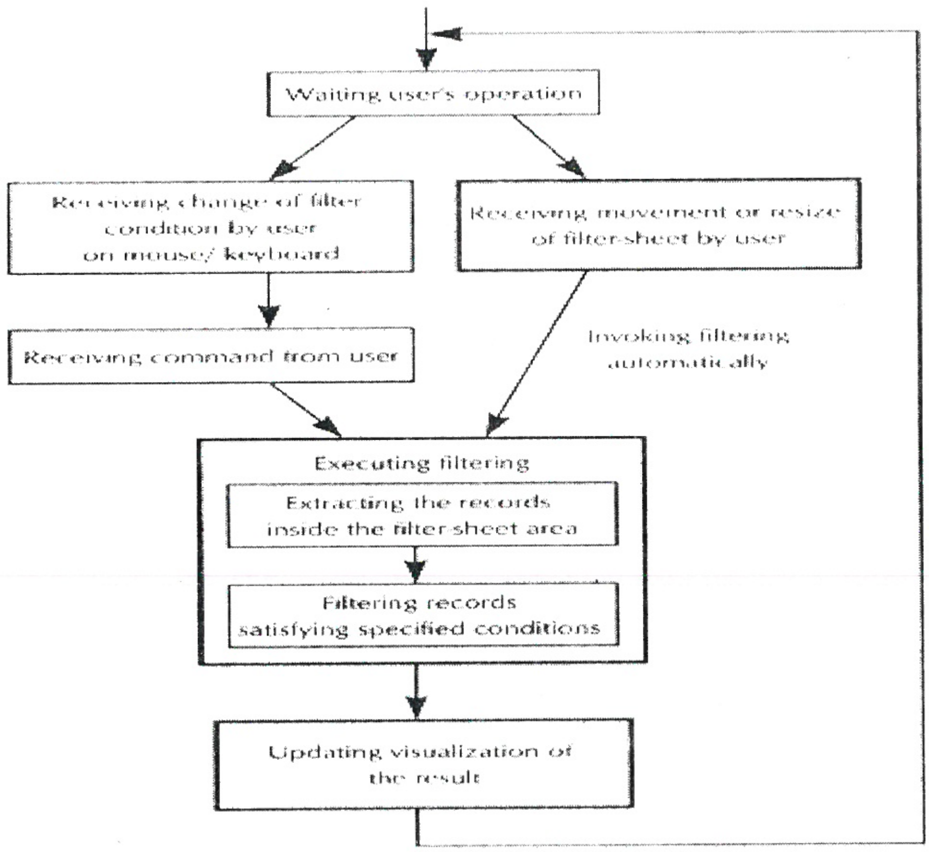

Figure 10. Flow of Dynamic Filtering Using Filter-Sheet

7. Conclusion

As basic functions of the visual query interface that supports users to search and manipulate records without knowing their data structure in DB, data visualization, image zooming, and dynamic filtering were proposed. Visualization helps users understand data easily from various viewpoints. The image zooming allows users to understand visualized data in various resolutions. The dynamic filtering supports users' trial-and-error in searching data, and its user interface is integrated with the visualization. The proposed query interface can be applied to supporting end-users with little knowledge of DB, or to supporting, for example, persons in the area marketing field.

As future work, several problems should be dealt with. At first, more consideration should be given the aspects of data visualization methods effective for data queries in practical use. Second, an evaluation of the usefulness and limitation is necessary especially in case of an increase in the number of object data to be manipulated.

REFERENCES

1. CHAMBERLIN, D.D., ASTRAHAN, M.M., ESWARAN, K.P., GRIFFITHS, P.P., LORIE, R.A., MEHL, J.W., REISNER, P. and WADE, B.W., **SEQUEL 2: Unified Approach To Data Definition, Manipulation, Control**, IBM JOURNAL OF RESEARCH AND DEVELOPMENT, Vol. 20, No. 6, 1976, pp. 560-575.

2. ZLOOF, M.M., **Query-By-Example**, Proceedings of the 1975 National Computer Conference, 1975, pp. 431-438.

3. SHU, N.C., **Visual Programming**, VAN NOSTRAND REINHOLD, 1988.

4. NORMAN, D.A.,. **The Psychology of Everyday Things**, BASIC BOOKS, 1988.

5. MIZUNO, H., MORI, Y., TANIGUCHI, Y. and TSUJI, H., **Data Queries Using Data Visualization Techniques**, Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics (SMC '97), 1997, pp. 2392-2396.

6. SMITH, D.C., IRBY, C., KIMBALL, R., VERPLANK, B. and HARSLEM, E., **Designing the Star User Interface,** BYTE, April, 1982, pp. 242-282.

7. SHNEIDERMAN, B., **Direct Manipulation: A Step Beyond Programming Languages**, IEEE COMPUTER, Vol. 16, No. 8, 1983, pp. 57-69.

8. BAECKER, R., SMALL, I. and MANDER, R., **Bringing Icons To Life**, Proceedings of the CHI '91 Conference, ACM PRESS, 1991, pp. 1-6.

9. CARD, S.K., ROBERTSON, G.G. and MACKINLAY, J.D., **The Information Visualizer, An Information Work-space**, Proceedings of the CHI '91 Conference, ACM PRESS, 1991, pp. 181-188.

10. MACKINLAY, J.D., ROBERTSON, G.G. and CARD, S.K., **The Perspective Wall: Detail and Context Smoothly Integrated**, Proceedings of the CHI '91 Conference, ACM PRESS, 1991, pp. 173-179.

11. ROBERTSON, G.G., MACKINLAY, J.D. and CARD, S.K., **Cone Tree: Animated 3D Visualizations of Hierarchical Information**, Proceedings of the CHI '91 Conference, ACM PRESS, 1991, pp. 189-194.

12. AHLBERG, C., WILLIAMSON, C. and SHNEIDERMAN, B., **Dynamic Queries for Information Exploration: An Implementation and Evaluation**, Proceedings of the CHI '92 Conference, ACM PRESS, 1992, pp. 619-626.