MINI - A 3D Mobile Image Browser with Multi-dimensional Datasets

Ai GOMI
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku
Tokyo 112-8610 Japan
gomiai@itolab.is.ocha.ac.jp

Takayuki ITOH
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku
Tokyo 112-8610 Japan
itot@is.ocha.ac.jp

ABSTRACT

Results of refined searches are often complicated because they have multi-dimensional attributes. Most of the existing retrieval systems display results in a grid or linear layout on small screens and do not visually represent the multi-dimensionality well. We think that multi-dimensional data visualization techniques can contribute to represent the distribution of the retrieval results based on multiple user-specified criteria and therefore assist the discovery of user-desired data items.

This paper presents MINI (Mobile Image Navigate Interface), a novel 3D visualization system for retrieval results adopted to run on mobile platforms such as smart phones. MINI allows users to interactively browse multi-dimensional datasets based on priority of data items calculated from multiple user criteria. It achieves the display of retrieved data items in the 3D space while avoiding overlaps and preserving adequate representation of the items' priorities. It also supports interactive orientation and provides a zooming user interface. We expect users can conveniently browse the retrieval results as well as are able to easily select the desired data items through the use of touch panel screens.

This paper also introduces the application of MINI in a recipe retrieval system, and the evaluation of effectiveness of the overlaps avoidance algorithm.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical User Interfaces (GUI)*

General Terms

Algorithm, Design, Experimentation.

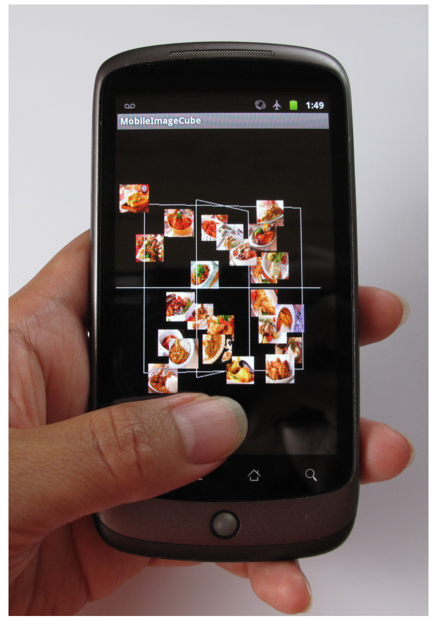

Figure 1: The presented 3D image browser MINI. The user is rotating the 3D space.

1. INTRODUCTION

The use of smart phones has been rapidly expanding throughout the world in recent years. They need to receive and display extensive amounts of complex data to end users provided through their Internet connection. It is expected that the overall amount of data used on smart phones continues to grow.

Information retrieval results from various data sources, such as recipes, books, online shopping items, and real estate often form complex datasets. They often have multiple attributes that users require to search for. However, it seems difficult to freely explore the multiple attribute spaces, because most existing retrieval systems on smart phones display the retrieval results in a basic grid or linear layout according to only one attribute on a rather small screen although the data has multiple attributes. We think that complex data visualization techniques used on smart phones can assist users to discover the desired information as quickly as possible.

Image-based representation of the retrieval results on a small screen is another challenge. The retrieval results from various data sources such as recipes, books, and online shopping items usually have associated photographs. We believe that the use of photographs or icons for the visualization of retrieval results is much more effective than using a textual representation. However, as the typical smart phones' screen estate is quite limited, it is challenging to visually represent densely-packed information in an appealing and easy-to-understand fashion. We present a novel technique for retrieval result navigation, MINI (Mobile Image Navigate Interface), for mobile platforms such as smart phones and tablet media players, which allow users to efficiently browse multi-dimensional dataset in a 3D space.

MINI prioritizes the data items in the retrieval result according to user's queries and displays the datasets staring from the upper-left corner of the display space in a 3D fashion in order of the priorities. The algorithm attempts to avoid overlaps as much as possible from the user's viewpoint, keeps the priorities, and retains the relationship of data in the 3D space. Since MINI enables users to efficiently navigate the multi-dimensional data in 3D space, users can explore the retrieval results based on their ranking from the users' multiple viewpoints. This is achieved through interactive operations, as shown in Figure 1. MINI achieves its user-friendliness through a highly interactive user interface. The system allows rotating the 3D space, while higher priority data is always placed towards the upper-felt corner of the display space as described in detail in Section 3.4. Focus zooming operation allows users to show detail information of each data. These mechanisms help users to quickly locate the desired and best-matching data on a small display.

2. RELATED WORK

There have been various previous works related to MINI. Image browsers and multi-dimensional visualization with a small screen are the research topics of the works most relevant to the presented browser. Our technique is a combination of these topics. This section introduces several related techniques.

2.1 Image Browsing

Many data browsing interfaces, such as data search engine Web sites, simply provide a set of data in a grid layout in the ranking order of a certain similarity criteria with respect to the query. These kinds of interfaces are not always effective for quickly finding all the desired data items, especially, in a small display environment such as smart phones. More sophisticated user interfaces for image browsing can be valuable. Kang et al. presented a technique for exploring personal image collections [9] which consists of various query interfaces, thumbnail and detail viewers; however, there would be too much information if the technique were used in conjunction with a small display.

Several browsing techniques focus on the layout of data so that they can finely represent content similarity among the data. We can embed data in a low-dimensional Euclidean space preserving the distances between pairs of dat, using, for instance, multi-dimensional scaling (MDS) [17]. Semantic Image Browser (SIB) [18] also applied MDS for the similarity-based layout of image thumbnails. Walter et al. presented Hyperbolic Image Browser [17] which scatters images onto a hyperbolic space applying MDS, and provides a novel focus+context user interface.

These techniques are good at representing distances among data; however, they often overlap many images onto each other on the display. Especially as many smart phones adopt touch panel displays in which case overlaps images pose a unique challenge in terms of allowing users to browse and select specific items on the often limited-sized displays. Tian attempted projecting the original high dimensional feature space onto the 2D screen based on Principle Component Analysis (PCA) [15], which solves the problem of overlaps data items on a 2D display. Another 2D-based approach to avoid overlaps is to apply grid-based layout of multi-dimensional datasets [3].

While many of image browsers display a set of images in 2D spaces, several techniques place them in 3D spaces. Gomi et al. [6] presented a personal photograph browser which places a set of hierarchically clustered images in a geo-time space and allows navigation through rotating and zooming operations. Miyazaki et al. presented an occlusion reduction technique, PileView, for a cityscape-style 3D visualization [13]. PileView places the set of piles by applying the hierarchical data visualization technique, and moves them to avoid the occlusion with users' interactive operation. However, the algorithm avoids the occlusion by expanding the bounds of the layout area beyond the initial area occupied before the technique was applied. For a small display however it is better to keep the layout area as small as possible even while avoiding overlaps. Our technique focuses on avoiding overlap of data items placed in a 3D space and displayed on a small display, all with interactive operations.

2.2 Multi-Dimensional Data Visualization

Information visualization techniques are suitable for data browsing because its main goals include the overview and interactive exploration of large complex datasets.

Scatterplots is one of the most popular techniques for multi-dimensional data visualization. Many scatterplots techniques directly allocate two or three dimensions in the visualization space. Rolling the Dices [4] is one of the novel techniques that introduces the interactive assignment of dimensions to scatterplots. Meanwhile, dimension reduction techniques are also popular ways to display multi-dimensional data.

While there are many well-known works on multi-dimensional data visualization, including Parallel Coordinates [8] and VisDB [11], we prefer scatterplots to display multi-dimensional datasets on smart phones, because we assume that most users are ordinary people who are more familiar with scatterplots than with other methods. We therefore present a scatterplot-based multi-dimensional data visualization technique for mobile platforms in this paper.

2.3 Data Visualization on Small Displays

There are several effective data placement techniques for small displays. Smart phones have strict restrictions in terms of display area and memory capacity compared with personal computers, and therefore interactive operations are key to understandably display the datasets on such platforms.

Several studies have attempted to apply data browsing strategies developed for desktop computers to small displays. For instance, Pocket Photomesa applies Treemaps to display photographs without overlaps on a cell phone [12]. Pocket Photomesa displays groups of photographs into rectangular regions, and allows users to navigate different levels of photographs using a zooming operation. Büring et al. implemented a scatterplot-based visualization technique which provides a smooth zooming user interface for small displays [1] . The techniques achieved to display 7,500 data items represented as dots on a small display. Mobile Liquid 2D Scatter Space also approaches a scatterplot interface for small displays [16]. It represents datasets of a movie database as size-varying circles. However, overlaps are still a problem of scatterplots. It is difficult to visibly recognize each data item because of too much overall density in spite of providing the zooming operation.

To visualize multi-dimensional data on small displays, MOVIH-IDS [7] applies a projection to capture and visualize UDP traffic datasets in a 3D space on mobile platforms. This approach focuses on displaying all data items for visual analysis, while our technique focuses on selecting data items from search retrieval results.

B. Karstens mentions that 2D visualization has an advantage of less consumption of computational resources compared with 3D visualization, and therefore they applied their 2D-based hierarchical visualization technique on a mobile device [10]. While this certainly was a critical challenge in past years; however, many current smart phones feature 3D graphical hardware accessed through powerful APIs,(e.g., OpenGL ES and Direct3D Mobile), and therefore we think it since became easier to run 3D-based software with smooth interactive operations on smart phones.

ZuiScat is a visualization system for querying large information on cell phones [2]. Retrieval results are presented in a dynamic scatterplot on a 2D space with a zooming user interface. Users can select attributes as query and assign them to axes. The system applies a clutter reduction technique and a logarithmic transformation to avoid overlaps of data. Our technique presented in this paper is also a visualization technique for retrieval results; however, our technique avoids overlaps as much as possible considering the priorities of the retrieval result in the 3D space.

3. MINI: AN IMAGE BROWSER IN 3D SPACES

This section presents a technical overview and details of MINI.

3.1 Technical Overview

Figure 2 shows the processing flow of data retrieval and layout processes. MINI supposes that retrieval results forming multi-dimensional datasets are provided from a database based on users' search criteria. We assume that the system first allows a user to input keywords or desired values of attributes. This section describes the i-th data item R_i as the following:

$$R_i = (w_{1i}, w_{2i}, ..., w_{m_j i}) \quad (1)$$

Here, w_{ji} is the j-th attribute in R_i, and m_i is the number of attributes used in R_i. We suppose that an attribute may be a single real value, or a set of keywords.

The system calculates the priority functions from the retrieval result, as shown in Figure 2 (1). We suppose functions to calculate the 3D position (x_i, y_i, z_i) of R_i as the following, as shown in Figure 2 (2):

$$x_i = f(w_{pi}), y_i = g(w_{qi}), z_i = h(w_{ri}) \quad (2)$$

where f, g, and h are placeholder names which are the functions to calculate the priority scores of the data items

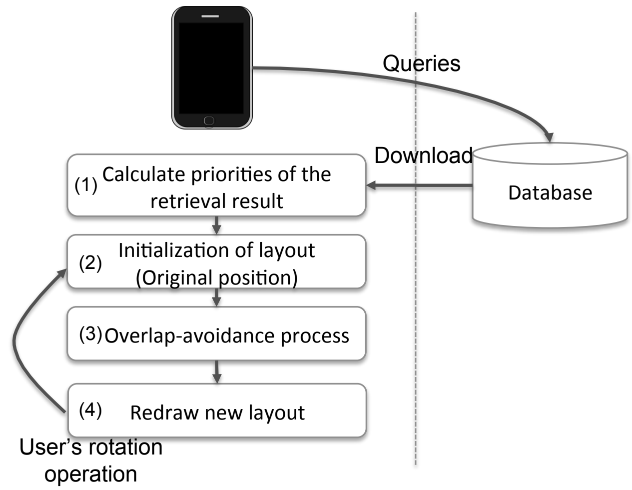

Figure 2: Processing flow of data retrieval and layout processes

based on their attributes. We suppose that the functions return positive values, and smaller values denote higher priority. For example, it is meaningful to develop a function that consumes the attribute of a single real value, and returns a smaller positive value if the attribute's real value is closer to the user-specified ideal value. Likewise, it is also meaningful to develop a function that consumes the attribute of a set of keywords, and returns a smaller positive value if the attribute keywords contain a larger number of the user-specified desired keywords. p, q, and r are integer values to specify the dimensions to be referred to calculate the priority of the data items. We can apply various attributes as well by replacing the functions, f, g, and h. This mechanism always places desired data items in the upper-left corner of the display area.

MINI then calculates the total priority of the i-th data item P_i by the following equation, the Euclidean distance from the origin:

$$P_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (3)$$

MINI places images associated to data items into a 3D space one by one, in the ascending order of the values P_i. It adjusts their positions from the original positions (x_i, y_i, z_i) to avoid overlaps from the user's viewpoint in step (3) as shown in Figure 2. This section presents a new algorithm for overlap avoidance, as described in detail in Section 3.3. MINI then moves and redraws the images from the original positions in an animated fashion as shown in step (4) of Figure 2.

MINI supports various user operations for image browsing including 3D rotation and zooming user interfaces. Rotational user interfaces allow rotating the 3D space along one of the axes. MINI inversely rotates one of the other axes during the rotation operation, so that the highest-priority images are always displayed in the upper-left corner of the display space.

3.2 Priorities

MINI allows users to input specific keywords or real values as well as to select three attributes out of the multiple

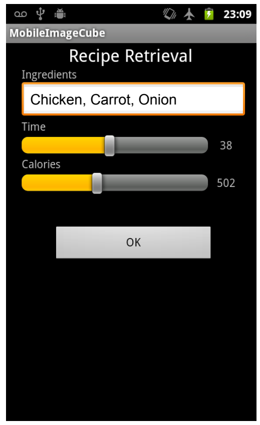

Figure 3: Query user interface. Users can enter keywords and various values for the query.

attributes offered on the first screen shown in Figure 3 In priorities based on keywords, L is the set of all keywords. W_i denotes the set of keywords contained in a data item R_i, where

$$W_i = \{w_{i1}, w_{i2}, ..., w_{im_i}\}, w_{ij} \in L \tag{4}$$

and m_i denotes the number of keywords in R_i. Also, let the set of user-selected keywords be S, where

$$S = \{s_1, s_2, ..., s_n\}, s_p \in L \tag{5}$$

and n denotes the number of user-selected keywords. If the set of keywords W_i contains all keywords in S, the data item R_i is returned as part of the retrieval results.

Our current implementation of MINI calculates the priority K_i for the results as follows:

$$K_i(k) = 1/\sum_{j=1}^{m_i} k_j \tag{6}$$

$$k_j = \begin{cases} j & (s_p = w_{ij}) \\ 0 & (s_p \neq w_{ij}) \end{cases} \tag{7}$$

This equation is used as the functions f, g, or h.

Our current implementation also calculates priorities from input real values V, by simply applying linear functions. Again, these linear functions are used as the functions f, g, or h.

3.3 Data Placement

MINI places data items in the 3D space according to the priorities calculated by the aforementioned step. The algorithm attempts to avoid overlaps data items as much as possible, while preserving the adequate representation of the priority of data items.

MINI calculates the positions of the data items in the order of the total priority P_i, where a smaller P_i denotes a higher priority, by utilizing the following algorithm:

1. Place the data item P_0, whose total priority is the highest. Here, P_0 is placed at the origin of the 3D coordinate system which is located at the upper-left end of the screen, as shown in Figure 4 (1).
2. Place the data item P_1, whose total priority is the second highest. Here, P_1 is placed at the intersection of the line l_1 with the circle c_1 as shown in Figure 4 (2). The line l_1 connects the original point of P_1 and the origin. The circle c_1 has the radius $r_1 = \alpha$, as shown in Figure 4 (2), where α is calculated by the sizes of images on the screen.
3. Place the data items P_i. Here, P_i is placed at the intersection of the line l_i with the circle c_i shown in Figure 4 (3). The line l_i connects the original point of P_i and the origin. The circle c_i has the radius $r_i = \sqrt{x_{i-1}^2 + y_{i-1}^2} + k$ $(k \leq \alpha)$, as shown in Figure 4 (3). k is a variable utilized to avoid overlaps with the already placed data items.

After the above process, MINI adjusts the positions of the data items by normalizing them based on the display size.

3.4 Swapping and Zooming User Interface

Figures 5 and 6 show the mechanism.

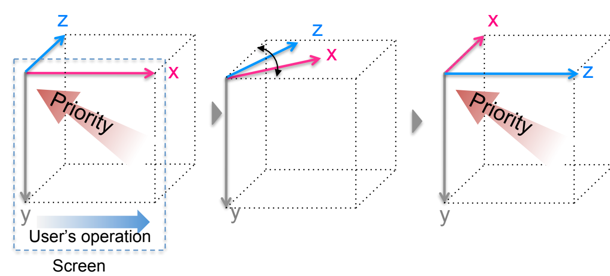

Figure 5: Swapping planes around the Y axis when the user swipes a finger from left to right on the display.

MINI supports various user operations for multi-dimensional data browsing in a 3D space. It features swapping view planes through rotations around the X- or Y-axis according to the users' operation on a touch panel screen. When the user swipes a finger from one side of the screen to the other, data items will rotate around the Y-axis. Likewise, when the user swipes a finger from the top to the bottom or vice-versa, data items will rotate around the X-axis. Moreover, during the swapping operation, one axis always rotates inversely to another so that the display always preserves the locations of high-priority data items towards the upper-left corner of the display space. For example, if a user browses the XY-plane and swaps the XY-plane for the YZ-plane, the Z-axis inversely rotates counter clock-wise, and consequently the YZ-plane is displayed instead of the XY-plane, all while the mechanism preserves the high priority data items in the upper-left corner of the display space, as shown in Figure 5. If a user browses the XY-plane and swaps the XY-plane for the XZ-plane, the Y-axis inversely rotates clock-wise, and consequently the ZX-plane is displayed instead of the XY-plane, as shown in Figure 6. The mechanism helps users to quickly discover the desired data items, because users only need to check the data items located in the the upper-left corner of the display space independent of whichever axes are selected.

MINI also supports a zooming operation and Fish-eye viewing technique [14], according to the users' operations

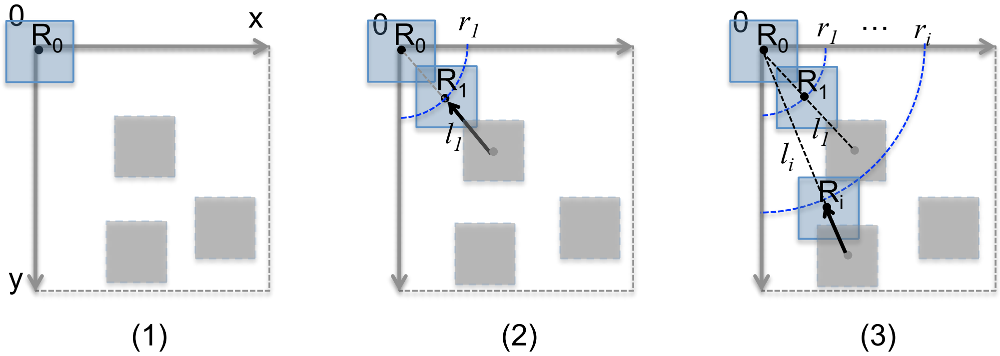

Figure 4: Algorithm of data placement.

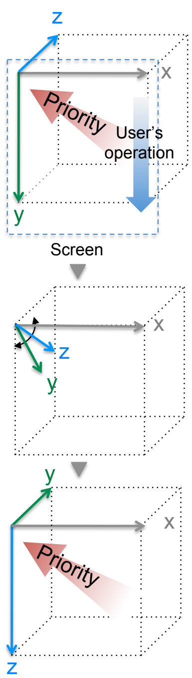

Figure 6: Swapping planes around the X axis when the user swipes a finger from top to bottom on the display.

of data item selection. When the user touches an image, the image gets enlarged while the other images get rearranged through a smooth animation, as shown in Figure 7. Also, when the user touches the image once more, detailed information associated with the touched image is shown so that the user can easily check and better understand the data item of interest.

4. APPLICATION TO THE VISUALIZATION OF RECIPE

We have applied MINI to a cooking recipe retrieval system to verify its effectiveness, since cooking recipes requires various attributes such as ingredients, category, cooking time, calories, amount of salt, and others to retrieve a recipe. Cooking is a necessary activity in our life; however, many

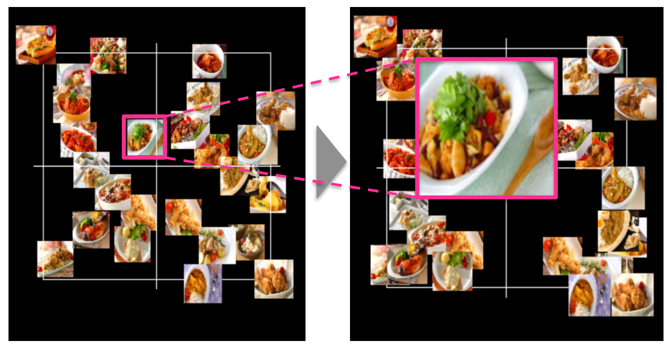

Figure 7: Data item selection. When a user touches an image (Left), it is enlarged (Right).

people who cook everyday feel bothered by the need for deciding what to prepare every time each and every day. Therefore, cooking recipe sites on the Web have been becoming popular in recent years because they can be helpful in the decision making process. In fact, about 55% of respondents in a survey about recipe retrieval from Netasia research answered that they use recipe retrieval systems on the Web, while 62% answered they use recipe books. This means that recipe retrieval systems are becoming common tools to retrieve recipes; however recipe books are still widely being used. Moreover, in the survey, about 51% of the respondents who use recipe retrieval systems on the Web answered that they have used recipe retrieval sites developed for mobile phones and tablet media players such as Apple's iPad. Such Web sites for mobile platforms are convenient because people can search recipes anywhere and anytime. According to the survey, the important factors for users in recipe retrieval systems on mobile platforms are the following:

1. User-friendly operation to easily obtain and understand retrieval results
2. Ingredients-based retrieval interface
3. Comprehensibility of food pictures on small screens

We have applied MINI to visualize recipe retrieval results to smoothly and quickly show the set of recommended recipes according to user's operations on a touch panel display, to achieve easy operation and understanding. We think people generally consider various attributes such as -perhaps most importantly- ingredients, but also category, cooking time, calories, and others, while retrieving a recipe. It is challenging to allow users to quickly get desired results of recipe retrieval if there are multiple requirements. Therefore, it is effective to display recipes by ranking according to user's queries. Although there are many categories of recipes, we selected three main attributes: ingredients, cooking time, and calories, all which are assigned to the three axes in the 3D space. Displaying pictures of recipes on the screen is also important for users while retrieving recipes because people can imagine what kind of cooking is to be done. Reflecting the above requirements, our study focuses on displaying the results of recipe searching refinement.

Our application assumes that a user first creates queries of desired ingredients as a set of keywords S, and cooking time V_1 and food calories V_2 as real values. Using these query attributes, MINI retrieves recipes considering the user's requirements from a recipe database. MINI then calculates the priority of each recipe in the retrieval result according to user's requirements for each of the attributes: ingredients, cooking time, and calories. MINI utilizes a 3D space with an orthogonal coordinates system to place the recipes; it assigns the X-axis to the priorities according to ingredients users selected in the queries, the Y-axis to cooking time, and the Z-axis to food calories, all while avoiding the overlap of recipe images and preserving the adequate representation of the priories. MINI provides orientation and zooming user interfaces which allow users to change their viewpoint in the 3D space by rotating planes and allows them to select a desired recipe. For example, if the display shows the XY-plane and a user navigates rightwards in the 3D space by operating the touch panel, the algorithm displays the YZ-plane instead of the XY-plane. Users can change attributes without changing the directions of an axis.

5. EVALUATION

We implemented the technique with Android OS 2.3.4, and tested on a Google Nexus One smartphone (Processor 1GHz, RAM 512MB, Display 3.7 Inch). We used the recipe site, E.recipe [5] published on the Web as a data source for our evaluation. Table 1 shows the sizes of three results of the retrieval in each plane. The table also shows the computation times for calculating positions of data items by the overlap-avoidance algorithm. The result demonstrates that the system calculates the positions within a reasonable time. When the rotation is conducted by a user, our technique applies the overlap-avoidance process and renews the layout of the data. During this time, the system interpolates positions before the layout renewal and positions after, and displays the movement as an animation.

Table 1: Dataset used in our experiments.

	The number of data items	Computation time (ms) XY	YZ	XZ
Dataset 1	25	22	21	26
Dataset 2	63	52	47	33
Dataset 3	189	108	95	131

5.1 overlaps rate

This section shows several examples of visualization results using the three datasets in Table 1 and evaluates the efficiency of our technique. Table 2 shows the contribution of our algorithm to avoid overlaps each data item as much as possible by comparing overlaps rates of before and after the placement.

The evaluation extracts overlapped regions and calculates their rates of areas $e_i = o_i/r_i$ for the i-th data item R_i. o_i is the area of the overlapped regions in the area of R_i, and r_i is the total area of R_i. Consequently, smaller e_i denotes the smaller overlap. Table 2 shows the maximum and average of e_i. Although the overlaps rate is increasing when the number of data items is increasing, the result demonstrates that the overlap-avoidance algorithm is much more effective. Especially, it becomes difficult for users to select a particular data item when a very large numbers of data items is placed on the display in which case many data items can completely overlap each other in the initial placement. The result demonstrates that the problem can be solved by applying our overlap-avoidance technique.

Figure 8 shows an example of parts of the animation while a user rotates the 3D space from XY-plane to YZ plane with the Dataset 2. Figure 8 (1) shows a result of the XY-plane before applying the overlap-avoidance algorithm so that we can observe that some data is overlaps on the screen. Figure 8 (2) shows a result of the XY-plane after applying the overlap-avoidance algorithm so that we can observe that data is overlap-avoidance as small as possible. Figure 8 (3) shows a rotation operation around the Y-axis. Figures 8 (4) and (5) show the YZ-plane before/after applying the overlap-avoidance algorithm. The figures show that MINI improves the visibility through use of the overlap-avoidance algorithm, with an easy selection operation on a touch panel.

Table 2: the maximum and average of e in each dataset.

Dataset	Plane		Before	After
Dataset 1	XY	Max	0.3055	0.0
		Average	0.3055	0.0
	YZ	Max	0.3055	0.0
		Average	0.254	0.0
	ZX	Max	0.3818	0.0
		Average	0.3230	0.0
Dataset 2	XY	Max	0.731	0.152
		Average	0.602	0.113
	YZ	Max	0.731	0.257
		Average	0.506	0.157
	ZX	Max	1.0	0.201
		Average	0.789	0.156
Dataset 3	XY	Max	0.911	0.561
		Average	0.697	0.405
	YZ	Max	0.911	0.542
		Average	0.471	0.374
	ZX	Max	1.0	0.412
		Average	1.0	0.372

5.2 Real-World Scenario

We visualized a set of recipes obtained as a retrieval result of the recipe searching refinement. We specified "Chicken",

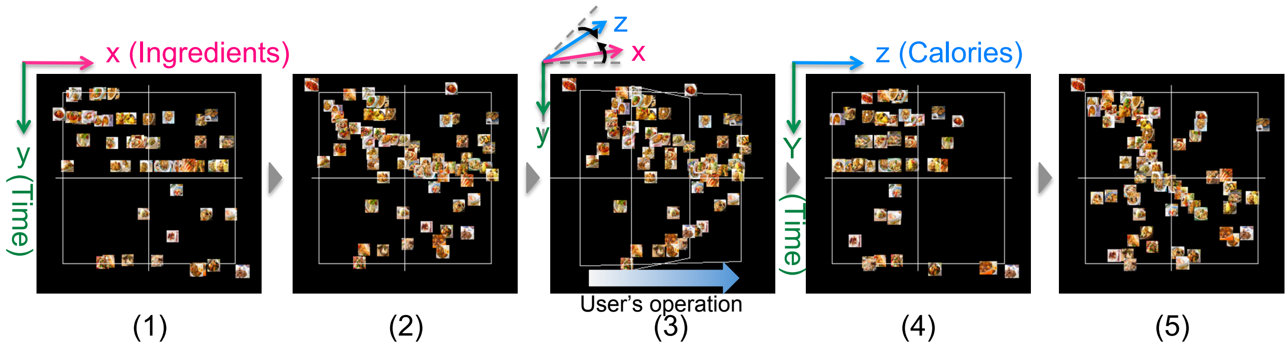

Figure 8: Results of a rotation operation by applying the Dataset 2. (1) Before applying the overlap-avoidance algorithm in the XY-plane. (2) After applying the overlap-avoidance algorithm in the XY-plane. (3) Rotation operation by a user. (4) Before applying the overlap-avoidance algorithm in the YZ-plane. (5) After applying the overlap-avoidance algorithm in the YZ-plane.

”Tomato”, and ”Potato” as ingredients, ”minimum” as cooking time, and ”minimum” as calories as the query to the recipe Web site. We obtained 25 results from the recipe database. Figure 9 shows the visualization of the results using the XY-plane and YZ-plane as viewpoints. Figure 9(Upper) shows that almost all of the recipe photographs of recipes using curry (enclosed by pink rectangles in the result) are placed on the lower right side of the display in the XY-plane. Even though the rotation is applied to display the YZ-plane, these recipe photographs are still placed on the lower right side of the display, as shown in Figure 9(Lower). From these results, we understood that the recipes using curry are poorly matched to the user's demand because of long cooking time, necessity of other ingredients, and higher calories. In addition, Figure 9(Lower) shows that the recipe photographs of foods involving frying (enclosed by blue rectangles) are gathered on the upper right side of the YZ-plane. We understood that fried food is easy to cook; however, it contains higher calories. Figure 9 (Lower) shows that several recipes that have a shorter cooking time and contain higher calories are using tomatoes (represented by red circles) but are not decorated in any other way. As this result highlights, displaying recipe photographs often helps users in arranging recipes too.

6. CONCLUSION AND FUTURE WORK

This paper presented MINI, a 3D image browser for small displays, which applies a data visualization technique of avoiding overlaps and keeping the priority of data resulted in a search refinement. Also, it verified the effectiveness of our technique in this paper by applying a recipe database as a search refinement. The application first calculates the priorities of recipes according to attributes of recipe data, ingredients, cooking time, and calories. It then places the recipes into a 3D space assigning the X-axis to the priority according to ingredients, the Y-axis to cooking time, and the Z-axis to calories, while applying the overlap-avoidance algorithm in the order of the total priority of retrieved recipes. The paper introduced several examples and experiments, and discussed the effectiveness of the application.

As the future works, we will investigate and discuss the following issues:

[**Experiments with very large datasets**] The limitation of the number of recognizable images on small displays is a serious issue. To be able to handle extremely large datasets, we could apply a clustering algorithm to gather similar data items and just display representative images of the clusters. It should be intuitive if MINI would display representative images of the various clusters while zooming out to browse the entire dataset, and each image within a particular cluster while zooming in.

[**Experiments with other search refinements**] We think that MINI can be applied not only to recipe datasets but also to other search refinements such as in the areas of online shopping, book search, and real estate. We would like to have additional experiments in such application fields.

7. REFERENCES

[1] T. Buring, J. Gerken, and H. Reiterer. User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):829–836, 2006.

[2] T. Buring and H. Reiterer. Zuiscat: querying and visualizing information spaces on personal digital assistants. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices services*, pages 29–136, 2005.

[3] R. D. de Pinho, M. C. F. de Oliveira, and A. de Andrade Lopes. An incremental space to visualize dynamic data sets. *Multimedia Tools and Applications*, 50(3):533–562, 2010.

[4] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1141–1148, 2008.

[5] Excite. E.recipe. http://erecipe.woman.excite.co.jp/.

[6] A. Gomi and T. Itoh. A personal photograph browser for life log analysis based on location, time, and person. In *ACM Symposium on Applied Computing*, pages 1250–1257, 2011.

[7] A. Herrero, E. Corchado, M. A. Pellicer, and A. Abraham. Movih-ids: a mobile visualization hybrid intrusion detection system. In *Neurocomputing*, volume 72, pages 2775–2784, 2009.

[8] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. Visualization 90*, pages 361–370, 1990.

[9] H. Kang and B. Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *IEEE International Conference on Multimedia and Expo 2000*, pages 1539–1542, 2000.

[10] B. Karstens, M. Kreuseler, and H. Schumann. Visualization of complex structures on mobile handhelds. In *Proc. International Workshop on Mobile Computing*, 2003.

[11] D. Keim and H.-P. Kriegel. avisdb: Database exploration using multidimensional visualization. In *IEEE Computer Graphics Applications*, pages 40–49, 1994.

[12] A. Khella and B. B. Bederson. Pocket photomesa: A zooming image browser for pda's. In *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, ACM Press*, pages 19–24, 2004.

[13] R. Miyazaki and T. Itoh. An occlusion-reduced 3d hierarchical data visualization technique. In *13th International Conference on Information Visualisation (IV09)*, pages 38–43, 2009.

[14] M. Sarkar and M. Brown. Graphical fish-eye views of graphs. *Fuman Factors in Computing Systems, CHI '92 Conference Proceedings, ACM Press*, pages 83–91, 1992.

[15] Q. Tian, B. Moghaddam, and T. S. Huang. Display optimization for image browsing. In *2nd International Workshop on Multimedia Databases and Image Communications*, 2001.

[16] C. Waldeck, D. Balfanz, C. Center, and G. ZGDV. Mobile liquid 2d scatter space (ml2dss). *In Proceedings of the eighth international conference on Information Visualization (IV'04)*, pages 494–498, 2004.

[17] J. A. Walter, D. Webling, K. Essig, and H. Ritter. Interactive hyperbolic image browsing - towards an integrated multimedia navigator. In *ACM SIGKDD*, pages 111–118, 2006.

[18] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *IEEE Visual Analytics in Science and Technology*, pages 191–198, 2006.

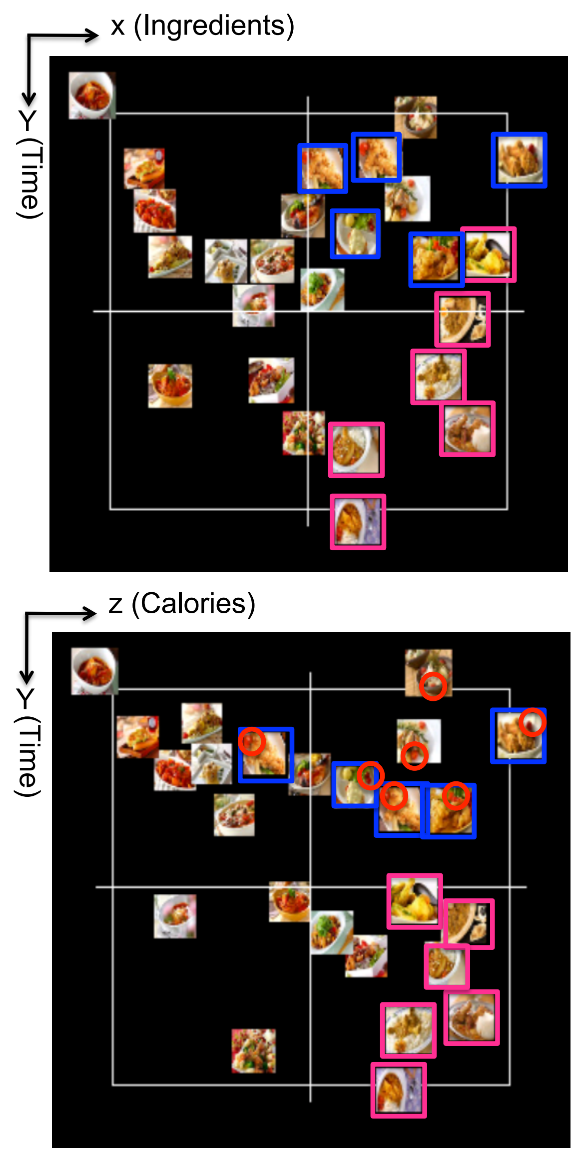

Figure 9: Results of Dataset 1. (Upper) XY plane. (Lower) YZ plane. Pink rectangles enclose curry recipes. Blue rectangles enclose recipes involving frying. Red circles denote the use of tomatoes.