Selecting One from Many: The Development of a Scalable Visualization Tool

Mark Apperley
The University of Waikato
Hamilton, New Zealand
m.apperley@cs.waikato.ac.nz

Robert Spence
Imperial College
London, England
r.spence@ic.ac.uk

Kent Wittenburg
Verizon Laboratories
Waltham, MA USA
kent.wittenburg@verizon.net

Abstract

This paper describes visualisation tools to support the task of selecting one object from a collection of many on the basis of its attribute values. For this frequently encountered task we identify a set of tools appropriate to a spectrum of collection sizes extending from hundreds of thousands to as few as ten or twenty. Although some of the tools have not previously been reported, and some have received only cursory attention in the literature, others are well known. This paper presents the tools in a coherent and consistent manner, showing relationships and progressions between them, identifying their principal attributes and relating them to the problem solver's cognitive task. We conclude with a proposal for integrating techniques within a single tool in order to deal with a continuum of working set sizes.

1. The problem

In the course of their daily life many people, both lay and professional, face a common generic task, namely, that of selecting one object from among many on the basis of that object's attributes. Often that task has to be performed without computational support: the choice of a tie to buy for a present, or the cheese for tonight's dinner table. In many other situations the potential for computational support is considerable. The layperson, for example, must choose a house to buy from among a realtor's collection on the basis of attributes such as price, number of bedrooms and journey time to work, notwithstanding additional considerations such as appearance. Similarly, the production manager of an electronics factory must place an order for a quantity of resistors, but only after exploring what is available in terms of supplier, delivery date, tolerance, reliability, price and other attributes. A system administrator may need to identify an intruder within the log files of a huge set of visitors to a web server, considering many attributes of the events logged. Again, other subjective considerations may apply. Indeed, it is the very presence of subjective considerations, the relevance of human experience and the often imprecise initial understanding of the problem that excludes the use of mathematical optimisation techniques and calls for the benefits afforded by visualisation tools [2].

1.1. Problem formation and solution

The invention and selection of an appropriate visualisation tool must take full account of the cognitive processes involved (after all, 'visualisation' is defined as the creation of a mental model of something). For example, it is almost always the case that the initial problem is imprecisely formulated. This is no fault of the problem holder, but simply characteristic of the way that most problems emerge. It is for this reason that – as a visit to any department store will confirm – a retailer will typically display a very wide range of goods to enable a buyer to form a mental model of what is available and thereby gradually decide what to purchase. In the same way a visualisation tool must make it possible for the user to gradually formulate their problem and then proceed to solve it. In fact, problem formulation and solution may well overlap, especially when planned activities mix with reactions to opportunities [12]. Thus, *learning* about the underlying data should be distinguished from *acting* upon the resulting mental model, for example to search for an object. It is also important to recognise the fact that a user will not only be assessing data in detail, but will also be looking for global effects, such as the decrease in house prices as one moves towards a particular area of a city.

1.2. The task and the tools

Mindful of the foregoing discussion, the overall objective of the tools to be described is to facilitate the following task:

"Given a collection of objects, each described by the values associated with a set of attributes, find the most acceptable such object or, perhaps, a small number of candidate objects worthy of more detailed consideration."

We emphasise the identification of a *set* of candidate objects in recognition of the fact that, as one whittles down the number of objects from perhaps many hundreds of thousands to ten or twenty, this process will occur gradually. We also remind the reader that the number of attributes can be high, commonly between 5 and 20. It will often be the case, moreover, that the user will – at least mentally – assign different degrees of importance to the various attributes. Also, the nature of typical data is such that a visualisation tool must be able to handle a mix of numerical, ordinal and categorical data.

The three visualisation tools described in Section 2 of this paper (*Dynamic Queries*, *Attribute Explorer* and *VisDB*), lend themselves especially well to the consideration of very large numbers of objects. In section 3 we describe *MultiNav*, *InfoZoom* and the *EZChooser*^{SM}, examples of tools most appropriate when several hundred or fewer items are under consideration.. Finally, in Section 4 we describe the *Neighbourhood Explorer*, particularly relevant when making a choice between a few objects – probably less than 30 – especially when non-quantifiable factors such as appearance are important. Obviously, many more relevant visualisation tools exist: necessarily we have made a selection to illustrate the factors arising from the cognitive activities of the user and the number of objects being considered. We also lay the groundwork for our proposal for a system that might integrate a subset of these techniques.

2. Large databases

2.1 Dynamic queries

Around 1990, the potential offered by a combination of enhanced computer power, inexpensive memory, improved display resolution and responsive interaction triggered the invention of an extensive range of information visualisation tools. An early, and seminal example was the concept of **dynamic queries** [15]. A simple illustration is offered in Figure 2.1: a person seeking to buy a house can place limits on various house attributes, whereupon those houses within the database are presented, and suitably colour-coded, within a map. Manual variation of any limit will cause the map display to be changed accordingly and instantaneously, allowing *sensitivity* (of appearance on the map) to various limits to be explored.

The dynamic queries visualisation tool is significant in many ways. First, it recognised that the house buyer may not have formulated their requirements with precision – indeed, that will be the most common situation. The dynamic queries tool allows the user to vary limits manually in order to *learn* (i.e., to form a cognitive model of) the database and, perhaps, to discover unexpected *opportunities*. That exploration may be cognitively

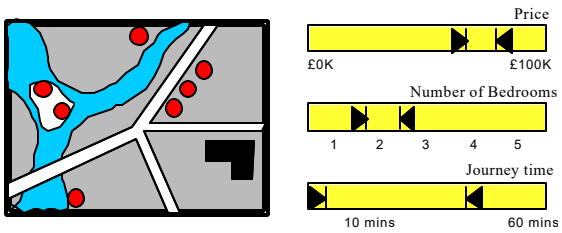

Figure 2.1 An example of dynamic queries

planned or opportunistically driven [12]. The mental model so formed may well be interpreted to suggest where to search next.

One of the many advantages of a tool such as the one shown in Figure 2.1 is its ability to handle a very large number of objects, and to provide the means whereby variation in attribute limits, followed by observation of the resulting change in the map display, can enhance the user's mental model of the database and support searching activity. The potential offered by the dynamic queries concept, as well as its utility, has been reflected in the success of commercially available visualisation tools such as Spotfire^{TM}[11].

Soon after the appearance and subsequent appreciation of the dynamic queries tool, desirable fundamental enhancements were identified leading to the invention of the Attribute Explorer, described immediately below.

2.2 The Attribute Explorer

The additional facility provided by the Attribute Explorer [10] can be illustrated by two examples based on the tool shown in Figure 2.1. In one, there are *no* houses that satisfy the specified attribute limits, and therefore no houses indicated on the map. Critically, no guidance is provided to suggest to the user that, for example, only a small amount of extra money need be offered to identify three houses worth consideration. In the other example, however many acceptable houses are shown on the map, it is necessary for the user to *manually adjust* the attribute limits to understand the influence of each limit on the availability and location of acceptable houses – in other words, to elicit sensitivity information.

These drawbacks can be overcome, and the huge potential of dynamic queries extended, by the **Attribute Explorer**. Rather than display only acceptable houses, valuable *context* and sensitivity information is provided by displaying *all* houses and colour-coding them according to the success with which they satisfy the attribute limits.

The essence of the Attribute Explorer visualisation tool is illustrated in Figure 2.2. We show, for illustration, three histograms related to three house attributes. As with the dynamic queries tool, limits have been placed on all attributes. Those houses satisfying all limits are colour-

Figure 2.2 An example of dynamic histograms in the Attribute Explorer

coded green in all histograms, whereas those which fail one, two or three limits are coded dark grey, light grey and white respectively. Such colour coding provides sensitivity information *without the need for manual exploration of the attribute limits*: a black house lying outside a limit will turn green if the limit is appropriately relaxed, *even in the situation where there are no green houses representing acceptable houses*. In some applications it would additionally be possible to select a dark grey area and thereby identify the *pairs* of limits responsible. A further advantage of the Attribute Explorer is that global relationships can easily be explored, for example by moving a range of Price up and down the scale and observing the general movement of green houses on the other histograms. In terms of communication, therefore, the Explorer conveys not only original data but its sensitivity to limits: previously, manual exploration was needed to elicit this sensitivity. Appreciation of global relationships can also be enhanced by encoding aggregate values such as average and variance directly and concisely on the scales (Figure 2.3).

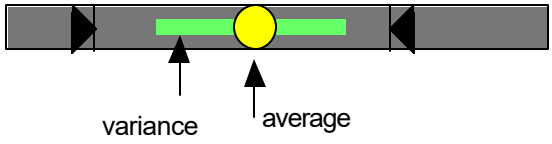

Figure 2.3 Encoding aggregate values on the scales

Many techniques can be employed to complement the Dynamic Query and Attribute Explorer tools. For example, since much can be learned from the appearance of a house, some means of presenting images on demand could well be introduced.

2.3 Exploiting every pixel – the VisDB visualisation tool.

In principle there is the potential for every pixel in a display to represent an object, and hence facilitate the visualisation of very large databases. Such potential was explored by Keim *et al* [5] in their creation of the **VisDB** visualisation tool. Like Dynamic Queries and the Attribute Explorer, VisDB allows attribute limits to be specified and adjusted and the results displayed graphically. But it also allows the user to indicate the level of importance to be attached to the satisfaction of each pair of limits.

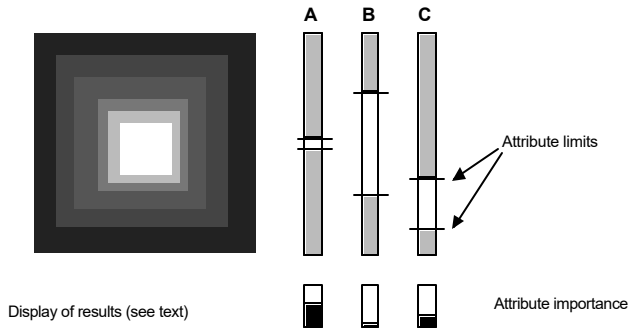

Figure 2.4 Techniques from VisDB shown schematically

The basic layout of the VisDB display is shown schematically in Figure 2.4, though without the benefit of colour which Keim *et al* have used to advantage to indicate the extent to which each object satisfies, with appropriate weighting, the attribute limits. Briefly, the three attribute scales on the right allow limits to be selected and, below them, the level of importance to be assigned. The (usually many) objects in the database are then sorted according to the extent to which they satisfy the attribute limits, and arranged in a spiral (shown on the left of Figure 2.4) with the most acceptable objects at the origin. Whereas different grey scales are used in the figure, Keim *et al* use colour coding, ranging from yellow (in the centre) to denote the satisfaction of all limits, towards purple at the extremity to indicate the least acceptable objects. The effect of varying limits and weights can be explored manually. For many other features of VisDB the reader is referred to Keim *et al* [5].

A major advantage of VisDB is its ability to handle a very large database. In the limit, if one pixel is assigned to each object then, depending upon the layout of the display, hundreds of thousands of objects can be handled. It differs from the Attribute Explorer in allowing the

importance of attribute limits to be taken into account. As with the Attribute Explorer, the VisDB tool also provides sensitivity information in the form of colour-coded scales.

As previously remarked, the three tools just described are best suited to the identification of a subset of objects worthy of more detailed consideration. Once several hundred or fewer objects have been identified as warranting examination in greater detail the visualisation tools described below are then appropriate.

3. The Bargram approach

The essence of the Bargram approach is illustrated in Figure 3.1 by reference to a single histogram (Figure 3.1a) of the Attribute Explorer. A one-dimensional bargram (Figure 3.1b) is derived from the two-dimensional histogram by 'tipping over' the columns of the histogram and laying them end-to-end, ignoring any null bins. The relative count in a bin is reflected in the bin's width in the bargram. Thus, the scale becomes one of rank or adjacency, rather than of relative value.

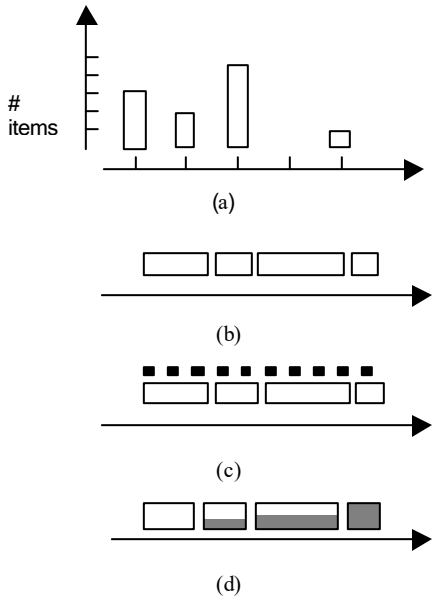

Figure 3.1 (a) A single histogram; (b) the corresponding bargram; (c) an item vector added; (d) some value information restored

Compared to histograms, bargrams carry less visual information about value distributions: for example, gaps are not shown (cf. Figure 3.1a and Figure 3.1b) and consequently outliers are not evident. Since items are simply ordered, there is no indication of whether adjacent items are close together or far apart with respect to a value scale. This loss of information is traded for a reduction in visual complexity as well as the advantage that each item in the set can be mapped to a simple linear position in a one-dimensional layout.

Since bargrams are related directly with item counts, it is possible to associate a parallel item vector with each bargram. Figure 3.1(c) shows bargrams with icons representing individuals above each row. (The item vectors can also be superimposed on the bargrams.) The same item will typically appear at a different horizontal position across different rows. An issue for this visualization is that at times orderings determined by the attributes may only be partial, but the visualization requires a strict ordering.

A variant on the bargram restores some information regarding value distributions, by reintroducing a second dimension as shown in Figure 3.1(d): shading or some form of line graph can be added to indicate relative value distributions across the bins. The addition of shading restores some of the information lost from the original histogram.

3.1 Visualisation tools

We are aware of three visualisation tools that incorporate interactive versions of parallel bargrams. They are InfoZoom [7], EZChooser [13], and MultiNav [6]. The interactive affordances that bargrams provide are evident in these tools.

InfoZoom [7], a data exploration tool, defaults to parallel bargrams in "overview" mode for larger set sizes and focus+context tables for smaller ones. The primary interactions supported in InfoZoom are simple selection and filtering, accomplished with a single mouse click on a value bin. The rendering advantages of parallel bargrams over table layouts at larger scales seem to be the primary motivation for their use in InfoZoom. Flipping between table and bargram views in InfoZoom immediately reveals that bargrams condense information relative to tables since same-valued cells are always adjoined in bargrams (and the labeling can thus be combined), but not with tables. Users can get a quick understanding of proportional distributions of values across multiple attributes with parallel bargrams that a table layout could only reveal after multiple sorting actions.

EZChooser [13] (Figure 3.2), a tool that has been deployed in online car shopping, utilizes parallel bargrams to support attribute exploration. Similarly to Attribute Explorer, users may select attribute values, rendered as buttons in EZChooser, to form dynamic queries. The results are reflected back onto the item vectors, as can be seen in Figure 3.2.

A second type of interaction allows the browsing and selection of individuals in the set and affords a multidimensional ranking view. The leftmost item among the cars at the bottom of the screen in Figure 3.2 has been "marked." This affords users a view in which a candidate

object can quickly be assessed with respect to its competitors in multiple dimensions. For instance, a user could quickly observe that a specific product might be at the upper end of a price scale but at the lower end of miles per gallon. Later versions of EZChooser allow marking of individuals at small scales and then viewing the markings at larger ones.

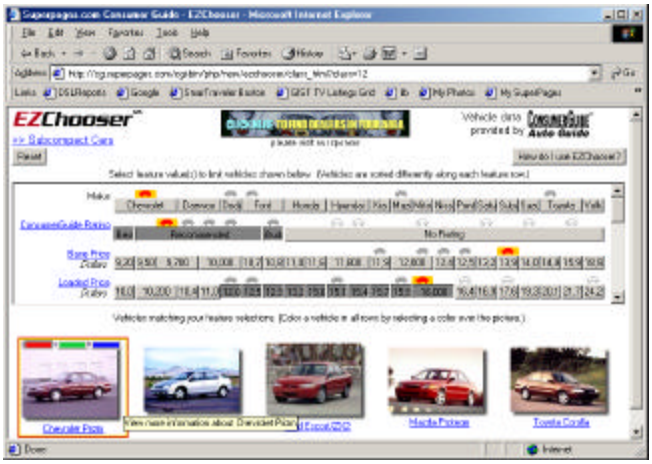

Figure 3.2: A screen dump of EZChooser[SM] . The user has reduced the working set "subcompacts" through attribute restrictions and is evaluating a candidate vehicle

MultiNav [6] utilizes another form of interaction afforded by bargrams coupled with item vectors. Consider a set of four histograms forming an Attribute Explorer display, with a single object highlighted in each dimension. In Figure 3.3(a), each of the four histograms has been "tipped over" as previously described, to form a bargram. The highlighting of the target object persists. It now becomes possible (Figure 3.3(b)) to translate the four bargram axes to make that object a common, albeit temporary, origin for all four axes.

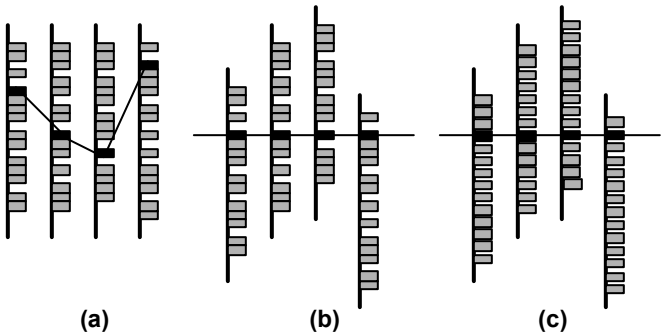

Figure 3.3 (a) An Attribute Explorer display converted to a set of bargrams; (b) the bargrams translated to bring the object of interest to a common origin; (c) the bargrams converted to item vectors

The primary browsing interaction in MultiNav allowed users to interact with the axes as sliders. In the so-called sliding dimension rod prototype, users could drag a dimension from side to side, changing the object of interest, with the resulting animation disclosing correlations, or the lack thereof, across the attributes.

4. Crossed item vectors

If the multidimensional space is sparsely populated, a refinement of the bargram approach is appropriate. In this case it is important to appreciate that the data is *discrete* in nature. Consider, again, the task of buying a house. Only a few of the candidate houses will be relevant to a particular buyer, and certain aspects of those houses, such as appearance, are non-quantifiable. The buyer has as much interest in making comparisons between candidate houses as in the individual attribute values. If the best match to their initial requirements is out of their price range, for example, the buyer will be interested in determining what compromises need to be made in order to achieve a more realistic goal. These compromises have to be *made in the context of available houses*.

4.1 The Neighbourhood Explorer

Consider again the multiple-attribute bargram display shown in Figure 3.3(b). If the number of objects (let us say houses in the current context) is relatively small, then we can dispense with the concept of value bins, and simply represent the individual houses on each axis as an item vector, as shown in Figure 3.3(c), remembering that the scale relates to *rank* rather than *value*.

Given a particular house of interest, comparison is likely to be made with those nearest on any given axis. For this reason the scale along each axis can be compressed beyond two or three data points by using distortion techniques [9]. This distortion now provides sufficient space on the axes for the houses near the reference house to be represented in iconic form, as suggested in Figure 4.1(a), acknowledging that the appearance of a house may be an important attribute in itself.

It is now apparent that a more useful representation of the relevant data is to combine all instances of the reference house into a single occurrence, displacing the axes sensibly as shown in Figure 4.1(b). This is the *Neighbourhood Display* for a particular reference house: it shows the nearest neighbours with respect to each attribute. Another house may appear more than once in this diagram, showing adjacency in more than one attribute dimension. In fact, the greater the number of instances of a particular house in the "focus" region, the greater its similarity to the reference house. The display also shows, along each axis, the ordinal position of the

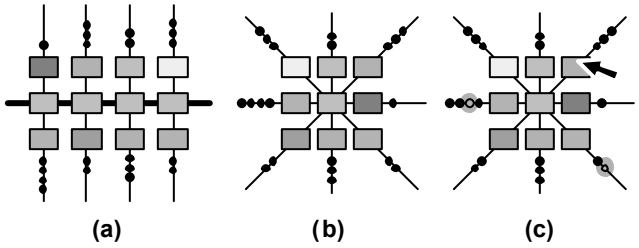

Figure 4.1 (a) The modified Attribute Explorer of Figure 3.3(c) further developed using distorted axes to allow houses near the origin to be represented as icons; (b) with all instances of the reference house combined and axes re-oriented; (c) with a new house of interest identified (note its presence on each axis)

reference house with respect to the corresponding attribute.

If the user's interest moves to a different house (Figure 4.1(c)) it will be necessary to generate a new Neighbourhood Display with this new reference house at its centre. This is achieved by "sliding" or translating each axis along its length to bring all instances of the new reference house to the centre.

Where houses have identical values for a given attribute (two houses having the same price, for example) they are still separated by unit distance along the axis, but their common value is indicated by a red or thickened line joining them (see later, in Figure 4.2).

4.2 Interaction with the Neighbourhood Explorer

An interactive implementation of the Neighbourhood Display provides a Neighbourhood Explorer [1]. The user is able to browse through the neighbourhood of an initial selection, looking for potential alternative choices. When a likely candidate is identified, selecting it with the mouse causes that house to become the centre of an updated neighbourhood display. Essentially, the user is exploring a multidimensional lattice of attribute space, where each object (house) is represented by a node. The neighbourhood Display is a localised view of this lattice, centred on the node of interest, and permitting duplication of other nodes in order to allow the view to be represented in two dimensions.

The power of the Neighbourhood Explorer is summarised in Figure 4.2. Here a group of houses is shown using four attribute axes. A fifth (horizontal) axis shows the browsing history; as a new house of interest is chosen, so this axis scrolls one unit to the left, carrying an instance of the previous target with it. In order to keep their goal in perspective, the user has been able to define ideal values for each attribute in advance, identifying a 'virtual house' represented by a "?" symbol on each axis.

Other interaction details shown in Figure 4.2 include the highlighting of all other instances of a house as the cursor moves over one of those instances: the display of an estate agent's description of a house if the cursor dwells on an instance of it: and the animated incremental translation of the axes when, to reinforce the preservation of context, a new house of interest is chosen as reference.

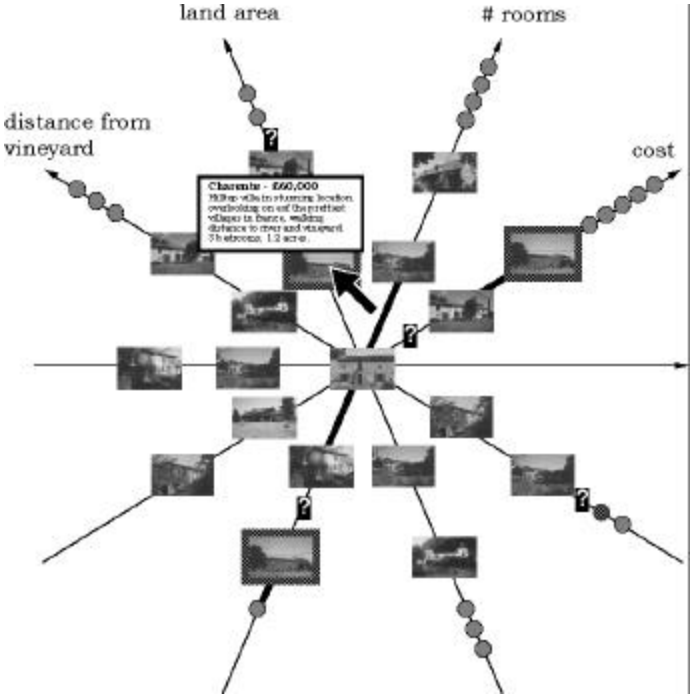

Figure 4.2 A Neighbourhood Explorer display for a group of 10 houses with four attributes

5. Conclusion

The visualization tools described above offer a 'palette' of techniques relevant to databases ranging in size from 10 or 20 objects to many hundreds of thousands, in the context of the reference task of selecting a single object to "best" satisfy a given set of attribute values. Each tool is relevant to a particular collection size through its facilitation of tasks appropriate to that size. These tasks range from the "whittling down" process applied to extremely large databases to the careful assessment, partly subjective, of a few objects. With large databases the user's focus tends to be on attributes in order to define a manageable subset moving, as the collection size becomes small, towards a focus on individual objects, their non-quantifiable attributes and their comparison with close neighbours. What is common to all these tools is their ability to support the formation of a good and appropriate cognitive model of the entire

dataset, and the provision of sensitivity information as guidance for further exploration and search.

Note that a user may need to "whittle down" the working set from thousands to tens as well as to subjectively assess just a few *over the course of a single session*. It is not sufficient to suggest that separate tools and techniques be used with given sized datasets, since the size of a dataset may change radically as a user carries out the task at hand. There are precedents for having the tools change as a result of the working set changing. InfoZoom [7] defaults to bargrams at large scales and tables at small ones. Query Previews [3] has two modes, one using dynamic histograms at large scales and another using dynamic querying at small ones. We look forward to further experiments that can integrate the set of visualization tools described in this paper. After all, choosing one from many is a generic task encountered often in everyday life and ever more often in everyday digital life.

6. References

[1] Apperley, M., Spence, R. and Gutwin, C. (1999) The Neighbourhood Explorer, *Working Paper 99/*15, Department of Computer Science, University of Waikato.

[2] Card, S.K., Mackinlay, J.D. and Shneiderman, B. (1999) *Readings in Information Visualization*, San Francisco, CA, Morgan Kaufman.

[3] Doan, K.,Plaisant, C., Shneiderman, B.(1996) Query Previews in Networked Information Systems, in *Proceedings of Advances in Digital Libraries*, ADL '96 (Washington DC, USA, May 1996) IEEE CS Press, 120-129.

[4] Inselberg, A.(1985) The Plane with Parallel Coordinates, *The Visual Computer*, **1**, pp.69-91.

[5] Keim, D.A., Kriegel, H-P. and Seidl, T. (1993) Visual Feedback in Querying Large Databases, IEEE, *Proceedings of Visualisation'93*, pp.158-165 and color plate CP-15.

[6] Lanning, T., Wittenburg, K., Heinrichs, M., Fyock, C. and Li, G. (2000) Multidimensional Information Visualization through Sliding rods, ACM, *Proceedings of Advanced Visual Interfaces (AVI'2000)*, pp.173-180.

[7] Lindner, H-G. (2000) Knowledge Reporting with InfoZoom, SAP Design Guild, Innovation Second Edition, 12/22/2000, http://www.sapdesignguild.org.

[8] Spence, R. (2001) *Information Visualization*, Addison-Wesley.

[9] Spence, R. and Apperley, M.D. (1982) Database navigation: an office environment for the professional, *Behaviour and Information Technology*, **1**, 1, pp. 43-54.

[10] Spence, R. and Tweedie, L. (1998) The Attribute Explorer: information synthesis via exploration, *Interacting with Computers*, **11**, pp.137-146.

[11] Spotfire – Welcome to Spotfire. http://www.spotfire.com.

[12] Tweedie, L. (1995) Interactive Visualisation Artefacts: how can abstractions inform design? In *People and Computers X, Proceedings of HCI'95*, pp.247-265.

[13] Verizon Superpages – EZChooser^{SM} help. http://www.superpages.com/EZ.

[14] Westphal, C. and Blaxton, T. (1998) *Data Mining Solutions*, Wiley, p 471.

[15] Williamson, C. and Shneiderman, B. (1992) The Dynamic Homefinder: evaluating dynamic queries in a real estate information exploration system, ACM, *Proceedings SIGIR '92*, pp.339-346.