Pervasive Information Visualization

Toward an Information Visualization Design Methodology for Multi-Device Co-located Synchronous Collaboration

Paul Craig, Xin Huang, Huayue Chen, Xi Wang, Shiyao Zhang
Dept. of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University
Suzhou, China
p.craig@xjtlu.edu.cn

***Abstract*—This paper looks at the potential of information visualization for multi-device co-located synchronous collaboration. That is, the capability of interactive graphical user interfaces to support face-to-face collaboration with networked devices such as large displays with motion recognition and mobile touchscreen devices. This is done by first considering how information visualization can be developed to fit the capabilities and limitations of different devices, then drafting a set of guidelines and recommendations for this type of application, and finally developing a simple prototype to demonstrate some of these guidelines in action. This shows us that that information visualization has the potential to support multi-device co-located synchronous collaborative and there is much promise for this type of software to be developed for a range of applications in the future.**

Keywords—Pervasive Computing, Information Visualization, Internet of Things

I. Introduction

There are many cases where people *need* to collaborate to achieve a particular objective. At other times collaboration isn't absolutely necessary, but it helps us to do a better job or have a better experience doing it [1]. Collaborative working is appropriate for a range of important human activities. Shopping, learning, sport, science, engineering and even entertainment can all benefit from face-to-face collaboration and to many of us *collaboration* is simply the most natural way to do things. In general, it's better if more than one person can work on the same problem together. This is collaboration.

Up until now, however, technological limitations have meant that the most common form of computer assisted working is that of one computer one user. This normally means that people work on their own. When people *do* work together they either use separate machines or one person has total control of the machine with the other looking over his or her shoulder. In either case they are not collaborating as effectively as they could do if they were able to spend more time face-to-face and focus more on each other rather than a computer screen [2, 3].

A new development that promises better support for co-located collaboration is the rise of ubiquitous computing [4-6] and mobile computing technologies [7] developed for the burgeoning smartphone market. Key components are mobility, touch control, improved display technologies and improved connectivity. Mobile devices mean that computing is no longer tied to a physical location so that people can move to meet each other and carry their data with them [7], large displays facilitate better face-to-face communication [8, 9], and better network connectivity means that data and resources are more easily transferred and shared between users [10, 11].

Despite this, interfaces for multi-device co-located collaboration are limited [2]. This can be attributed to device limitations such as screen space and input peripherals [12], social factors [13, 14], and the complication that every aspect of an interface (interaction, security, display etc.) has to be operable by multiple users at the same time [14]. Natural sharing of control and display space together over multiple devices is an important consideration that has not been addressed by current research [15, 16]. On the other hand we have seen that information visualization [17] techniques show great promise for overcoming device limitations [10, 18, 19] and, in other studies, managing collaborative working with multiple users [12, 20-22]. HCI research has also demonstrated techniques for multi-user control on table-top displays [23] and single-user control of applications on multiple devices [24, 25].

This paper investigates the feasibility of adapting information visualization techniques for co-located synchronous collaboration by considering how different aspects of information visualization design can be applied in a multi-device multi-user environment. This forms the basis for a set of draft guidelines for multi-device co-located synchronous collaborative visualization interface design. These guidelines are applied to the design of a prototype application to test how they can be applied in practice.

II. Requirements Analysis

Our initial case study for multi-device co-located synchronous collaboration, the *HotelFinder* application, allows tourists to look for a hotel together. This is based on the seminal *HomeFinder* application [26] where users can find a new house by looking at a starfield display with attributes such as the number of rooms, location, price and area coded into visual variables like color, shape and position. Our application is a multi-user multi-device variation on this theme which helps people to find hotel rooms rather houses or apartments.

If you travel with other people finding a hotel could clearly benefit from being a collaborative group activity with different users being able to make decisions together in the same place at the same time. Users are also likely to want to take data away with them in order to find their chosen hotels when they are on holiday, or even decide between different candidate hotels or review their decisions on the move. Hence, we could imagine a multi-device multi-user environment where there is a single big-screen for group activities and mobile devices for members of the group to review their decisions on the move.

An initial focus group with a group of potential users allowed us to gain some insights into user expectations of how a multi-device collaborative information visualization like this should work. This was undertaken with a small group of six Chinese university students. The students included two males and four females aged between 21 and 23. They considered themselves reasonably tech-savvy with experience of mobile devices and the use of websites for tasks such as shopping, paying bills and, most importantly, finding hotel accommodation.

The initial impression of the students had of our proposal was that information visualization software is complex and only really appropriate for scientific applications. This attitude changed however when we showed the students some more accessible examples of information visualization applications such as the Mexican History Explorer [27], used to explore web search results using coordinated map and time-line views, and the Dialogue Explorer [28], used to look at human dialogue. This led them to think that an information visualization could be quite useful for finding a hotel. They were particularly encouraged by the idea that they might be able to have a more sociable computing experience and discover a more informed way to make group decisions.

In order to have an idea of how our case-study application might work we asked the users to walk through an ideal scenario whereby they could use a multi-device application to realize their objective of finding a hotel with their friends. This scenario involved the following steps. First of all, the students envisioned meeting together to see what types of hotels were available, having a discussion of what factors were important and then deciding on which type of hotel they wanted. Important factors would be the price, rating, the proximity of amenities, and the distance from amenities or sightseeing locations. Their initial discussion of these factors would allow them to compile a list of hotels in different areas that they might consider as suitable accommodation. Then, they would decide on a hotel and book it or, if their holiday plans were less concrete and included for example a day trip they might possibly take to an outlying location, they would take the hotel details with them to make a booking once they made a proper decision about where they wanted to go.

In the cases where the group initially met together they felt it would be useful to have a large display, a desktop-pc or at the very least a laptop-pc to help them make their decisions. When the students where in their destination a mobile-phone or a tablet would be more appropriate.

We also asked the students what would be likely to be the most important factors affecting their experience with this type of system. The students agreed that the overall most important factors would be learnability and ease of use. The students didn’t want to have to learn to use the system or have to operate too many interface components. It was considered that these factors would be especially important when multiple users worked together on the same interface, as individual users didn’t want to be made to feel foolish if they couldn’t operate the interface or be distracted by too many options while trying to communicate with the rest of the group. They also wanted to be able to connect devices with the minimum amount of interaction and for them to connect automatically if possible. These feelings are consistent with the findings of other researchers who specify that interaction with this type of interface should be fluid and seamless [29] and that mobile interfaces should feel natural and focus on interaction with the data rather than including too many menus and options [30].

III. DESIGN RATIONALE

Having decided to focus our investigation on the design of an application to help a group of users find a hotel with a large display and several smartphones or equivalent mobile devices, the next stage was to relate user requirements to individual design decisions. These related to the different visual mappings and view transformation in each display, and particular features of the interaction and display for different devices.

A. *Visual Mapping of the data in each display*

The visual mapping of an information visualization determines how data tables are mapped to visual variables such as spatial displacement, size, shape and color. This transforms raw or processed data to visual structures and determines the form of the visualization. For example, two alternative mappings of the same data could encode different scatterplot axes or make the same data appear in the form of a line-chart.

There are a number of reasons why we might have different visual mapping in the different displays of a multi-device multi-user environment. Firstly, different mappings could be complimentary showing different aspects of the data. For example, in the case of our *HotelFinder* application, users could easily select hotels on their mobile according to their price and the rating using a scatterplot, and use a map on a large display to see the location of their selected hotels. Selections could be coordinated making the displays act like the multiple coordinated views [31] of a normal visualization. Another possibility is that different mappings are more appropriate for different devices due to factors such as limited screen space or different input modalities.

A problem with changing mappings between displays is the additional cognitive load for the user who has to shift their attention between these mappings and look at alternative projections of the same data. So, we should make sure that the benefits of moving between mappings are worth the effort. In the case where users have to move their attention between complimentary views of the data we can normally see a clear advantage in having the two different mappings. For example, moving between a map and scatter-plot showing house prices and location gives the user a sense of both house-value and geographical context when it would be difficult to communicate both of these qualities in a single view. In this type of case it’s likely that the additional effort from the user is rewarded through additional insight into the data. Where mappings are changed due to device limitations, the advantage is less obvious. This makes it appropriate to keep equivalent mappings (i.e. mappings that try to show the same thing) as consistent as possible on different displays to minimize the cognitive load on the user who has to move their attention between these displays.

In the case of the *HotelFinder* application we can see a case for both complimentary and consistent visual mappings. As discussed before, users can select hotels on their mobile according to the price and the number of bedrooms using a scatterplot view and use a map on a large display that reveals the location of the apartments. The large display could also contain a bigger version of the scatterplot for people viewing without a mobile device and it would be useful if the map could also be accessed from the mobile devices when the users move away from the large display to go to out and view the actual properties. The important thing is that *equivalent* views on different devices are as consistent as possible to avoid unnecessary additional cognitive load.

Other situations where alternative mappings might be appropriate are those where users have different roles more conducive to different mappings, or there is some process where users need to assume different roles or perform different types of task. For example, a group of ecologists may need one visual mapping to help them identify specimens in the wild and another mapping to assess the data with their colleagues. The same rule applies here. Complimentary views can differ but equivalent views should be as consistent as possible.

B. View Transformations in each display

Next we consider how to coordinate view transformations in each device display. View transformations, also known as data brushes, are individual selections made on the visual structures of the data. For example, if one of the students in our *HotelFinder* case-study wants to highlight or select a particularly interesting group of hotels and show this to the rest of the group, this selection is a view transform. The result of a data-brush is normally either what we can call a *soft-selection* to highlight or label the data momentarily or what we will call a *hard-selection* which filters or highlights the data until another selection is made. On a desktop PC application *soft-selections* are normally made hovering the mouse pointer while *hard-selections* are made by clicking or clicking and dragging the mouse pointer.

The question is how to coordinate selections on different device displays. Different options are:

1) Independent displays: This option means that each device display responds independently to user interaction through it’s own interface and there is no coordination between displays. This reduces the possibility of conflict between user-selections since users need to actually be interacting with the same device for any conflict to occur. The problem is that there is less capacity for sharing and if a user wants share a selection made on their own device, they would need to repeat the selection on the main display.

2) Complete coordination: This means that any selection made by any user appears on all devices imediately. The problem with this approach is the capacity for conflict between user-selections. For example, if a user is in the middle of forming a selection and another user makes a selection this could be distracting or confusing, especially in the case of a hard selection where the elements they select could unexpectedly disapear.

3) Coordination on main display: This means that selections made on mobile devices are sent to the main device but not visa versa. In this case any conflicts between user selections will affect only the main device and not personal mobile devices. Users can share selections on the main display but cannot view another users selections on their own personal mobile device.

4) Commands to send and retrieve: In this configuration users can choose to send selections to the main device or retrieve selections from the main device to their personal devices. This adds an additional step for each user if they want to share a selection but offers more control over what is shared and what is kept private.

The preferred option for the *HotelFinder* application would be for coordination on the main display for soft-selections and commands to send and receive for hard selections. This way users could immediately highlight hotels on the main display when in the discussion stage of finding a hotel, and send and receive lists of hotels when they are closer to making a decision. At this point hotel lists could be stored on the users’ mobile devices in preparation for actually visiting the hotel on holiday.

C. Interaction Style

Our next consideration is the the actual interaction mechanism on each of our different devices. Here we consider that since equivalent visual mappings should be consistant between displays to reduce the cognitive load of operating the display, interaction should also be consistant or that we should at least have equivalent interactions that map easily from one type of display to another.

Our extensive, but by no means exhaustive, list of user interactions are considered as follows:

1) Inspection: This is the most basic action on desktop PCs, equivelent to hovering the mouse-cursor to momentarily reveal some details about objects under the mouse cursor, and an important action for most visualisation applications [32].

The closest equivalent to this action on touch-screen devices is allowing the user to move their finger across the touch screen to move a virtual cursor. This is, however, problematic as the user’s finger is likely to obscure whatever is being inspected. Moreover, pressing and moving your finger is more equivalent to a drag action where, on a desktop computer, a mouse button is pressed while the mouse is moved. If a movement is made without pressing on the touch-screen it will not be detected. On larger touch screen devices longer swipes can also cause fatigue.

Large displays with motion detection devices can emulate this type of action by sensing the position of the users hand in front of the display and positioning a cursor according to this position. Unfortunately, prolonged interaction of this type can cause fatigue and the lack of physical contact make it difficult for the user to maintain the position of their hand and it is difficult to fix the position of the cursor or attain any real degree of accuracy. Moreover, movement is always monitored and it's impossible for the system to distinguish between intentional interaction and unintentional or involuntary actions such as the user scratching their nose or rubbing their eye. These factors could make the experience of inspection type navigation using motion sensing devices a less than satisfactory experience for most users if they are not accounted for in the interface design.

2) Selecting an object: This is another fundimental action for most information visualisation applications. On a desktop PC this action is normally realised by pressing the left mouse button while the mouse cursor is positioned over the object.

The equivalent action on a touchscreen device is to simply press on the object. The big difference with this action is that is lot less accurate than a mouse selection. This is because touchscreen devices have no way to sense the position of your hand before it presses on the screen and no floating cursor meaning the user cant line-up the cursor before making their selection. This problem is exacerbated by the small amount of screen space on mobile devices.

Another possible equivalent for this type of selection on touchscreen devices is for users to move a cursor by pressing and moving their finger until they are sure the object they want is inspected then waiting until a timer elapses to select the object. While this allows the user to be more accurate and has the advantage of allowing inspection type interaction, it adds a frustrating delay to each interaction and is unlikely to be conducive to what we might think of as the *fluid* style of interaction necessary for a satisfactory user experience. Another option is to highlight the inspection and wait for the user to remove their finger before the selection is made (as in the case of many mobile keyboards). The problem with this option is that it is counter-intuitive and makes it difficult for the user to cancel a selection if, for example, they don't find the element they want.

Point-and-click style selection is even more problematic with motions sensors and large displays. Unlike desktop or mobile interfaces there is no equivalent to a mouse-click or a press for motion sensor displays. This means that the user typically has to move their hand into position and make a gesture or wait until a timer elapses in order to make a selection. This can make selection frustrating and impractical for anything but the most basic of interfaces.

3) Selecting an area: This action is realised using a click-and-drag action on a desktop PC. The equivalent action on a touch-screen device is to press and drag your finger. This is simple on a smaller mobile device but larger motions on large touch-screens can cause fatigue. This action is also impractical on large-displays with motion detection because there is no obvious equivalent of a mouse-press or mouse-release to initiate and terminate the action.

4) Moving an object: Moving an object has the same click-and-drag action on a desktop pc as selecting an area. This has a similar action on touch-screen devices and suffers from the aforementioned problems of fatigue on larger touchscreen displays.

5) Entering text: On a desktop PC text is entered via the keyboard. Most touch-screen device operating systems have a virtual keyboard but these are less efficient that real physical keyboards due to the lack of tacktile feedback. Future devices are likely to rely on speech-to-text for text input as this type of technology becomes more effective.

6) Shortcuts: On a desktop PC the keyboard is also used for specialised shortcut keys or key combinations that triger specialised actions. These can be substited with virtual buttons on a touchscreen interface and gestures on a system that relies on motion detection.

The first thing we notice about this list of interactions are the limitations of larger touch-displays or motion detection devices that might be used to make large displays interactive. While some of these limitations might be partially resolved by minor technological advances in the near futures, other limitations such as the impracticality of large sweeping touch gestures or the accuracy of motion detected selections depend on the user and are unlikely to be resolved without completely rethinking the technology. This should not however be a problem with our proposed solution since visual mappings are coordinated on a main display and the main display can be controlled from any connected mobile device. If users are to be able to interact on the large-screen display it should be limited to interactions that are natural for the device.

Another potential problem is the inability of a touch screen to differentiate between inspection and selection type actions where dragging (with the mouse button pressed) is normally distinguishable from hovering (without pressing the mouse button) on a desktop pc. A possible solution is to use a toolbox metaphor where the user can press buttons in the interface to select different tools that operate in either inspect or selection mode.

Other problems with the touch screen display are occlusion by the user's finger and the inaccuracy of selections. The occlusion problem can be solved by labelling that shows selection details away from the users finger. Inspection details can also persist for a time after the user lifts their finger. The problem of accuracy can be resolved by making selection targets larger or using selection techniques that do not rely on accuracy and select multiple objects around the point of selection. Given these criteria, excentric labelling [33] would appear to be an ideal technique for mobile visualization since it allows us to select an area of objects and could be easily modified to shift labels away from the area hidden by the user's digit. Selection techniques that do not depend on accuracy are also appropriate for large displays that rely on motion detection for interaction.

D. Display style

As we have seen in the previous section, some aspects of the display design for multi-device collaborative visualizations depend on how we interact with those systems. For example, touches are inaccurate so we need larger targets for selections or large area selection methods like excentric labelling [33]. We also need to ensure that important information is not obscured by our finger when we inspect the data and we can't rely on mouse buttons being pressed to differentiate between hover and drag actions so we need additional buttons or some other way to do this.

Other aspects of our display depend solely on the display size and how we can best convey information in small or large displays. A basic trade-off already established is that the quantity of information displayed should be proportional to the amount of available screen-space [29] and the time we have to interact with and look at the data. Hence, to convey the same amount of information in a smaller interface we need a more interactive interface and the user has to spend more time interacting with the visualization. Conversely, on a larger display we can display more on each screen and the user can rely on simple visual inspection and will not have to spend so much time interacting with the data. If we think of an information visualization as consisting of different levels, starting with the overview and being filtered down successive levels to the final detail view of a single item, a mobile visualization simply has more layers than its large screen counterpart.

This makes the need for fluid interaction even more pressing on mobile displays. If we have additional interaction steps on a mobile device and these steps are not natural then by the time a user navigates from one view to another they may lose something of the sense of the initial view in the process of the interaction. For example, in the process of navigating to a detail view the user may lose their comprehension of the overview. This could also be problematic when comparing elements in different views or relating between different views that contain the same element. Both of these operations would be relatively simple using juxtaposed coordinated views but these are only really practical in larger displays. Another strategy to partially resolve this problem is to use animation to smooth the transition between views. Careful planning of how users can naturally navigate through different views of the data could also be useful.

Configuring the layout of individual views is also important. If we consider how HTML renderers scale web-pages, by keeping certain elements such as the size of the text fixed and scaling all other elements to fit these restrictions to make optimal use of the available display space, we can apply a similar logic to the scaling of information visualization interfaces. Items with a fixed minimum size for any given device are text (which has to be readable) and the target for user selections. Other variables that need to fit around these variables are the number of elements or aggregates displayed on the screen. If the information the user wants cannot be displayed they need to be able click on an aggregate to zoom-in to the visualization. So, a mobile view will typically have a greater degree of data aggregation in order to display more elements effectively on a smaller screen and rely more on user interaction to explore the data whereas larger displays can afford to show more of the data at any one time without aggregation or the necessity for animation.

E. Other issues

Other important issues for co-located synchronous collaboration are to do with security, privacy and the subtleties of social interaction. Multi-device multi-user applications necessarily means more interconnected devices belonging to more users. The problem with having more things interconnected and accessible is that there are more data available with more sensors and more opportunities for malicious attacks where the network can be hijacked or sensitive personal data can be leaked due to inadequate security protection. Indeed, if users interact with a display in a public space then an invasion of privacy might even occur through the simple act of eavesdropping where someone simply reads sensitive data from the display. This raises some difficult questions related to privacy. When is data private or public? What do we think of people reading our data even if it isn't private? Who do we want to look at our data? What about personal space and eavesdropping? While none of the users in our initial focus group raised concerns over privacy we could imagine their attitude might change if were someone to be looking in on their personal selections or they were involved in a more sensitive activity such as exchanging image files.

Another issue related to security and privacy is trust. Would people trust a connection with a public display? Before answering this we need to consider how people would actually connect to such an interface. Would it be automatic or would a user need to give their permission? Do they connect one time or every time they come into range? According to the vision of the internet-of-things, connection would be largely automatic but many questions of security and user acceptance are as-yet unresolved.

We also need to consider the additional personal question of how people work together using co-located multi-device multi-user interfaces. While we know that there is certainly the potential for these types of interface to foster productive collaboration [34-36] and many of our design decisions have been made in an effort to try and encourage this type of working. Social interaction is a complex and unpredictable phenomena that really needs to be observed to be properly understood. This makes it important for us to plan for the development of higher-fidelity prototypes and more advanced user studies as we continue this investigation.

IV. Design Decisions

From our analysis of the *HotelFinder* case-study, including our study of previous work, requirements analysis and the development of our design rationale, we can summarize our design decisions as follows:

1) Interaction should be fluid. The interface should askew menus, buttons and other widgets in favour of interactive representations of the data [30]. This is particularly important for co-located colaboration as any difficulty the user experiences in operating the system may inhibit communication between users and prevent the process of productive collaboration developing in a natural way.

2) Similar views should use consistant visual mappings and equivalent interactions. Where different displays show the same facet of the data they should be consistant wherever possible so as to minimise the effort made by the user to adjust to different visual mappings and different forms of interaction.

Where the same type of interaction is not possible on different display types an effort should be made to use equivalent interactions on different devices so that interaction is more natural when moving between devices. Efforts should also be made to design scalable visualizations that are easily adapted for different sorts of device.

3) Avoid conflict between users by differentiating between soft and hard selections. Coordinating between views on different displays is more complicated than coordination on a single-user single-device visualisation since the actions of one user have the potential to interfere with the actions of another user. To avoid these conflicts we can differentiate between a soft-selection to highlight or label the data momentarily and a hard-selection which filters or highlights the data. Soft-selections can be easily coordinated without much potential for conflict. More care needs to be taken over the coordination of hard-selections which have the potential to over-write the selections of other users and cause conflict.

4) Stick to functionality that is appropriate for the type of device. Different devices have their own advantages and limitations. However, multi-device systems have the advantage that no single device needs to include the functionality of the entire system and different aspects of the system functionality can be restricted to devices where that functionality is appropriate. Moreover, functionality can be distributed among devices to take advantage of their capabilities to optimise the user experience.

For an application designed to work on several small mobile devices and a single large display with motion sensor input, these general guidelines translate to the following recommendations:

1) Use techniques that make more efficient use of available screen space or do not require accurate selections. Techniques such as excentric labelling and space filling layouts fulfil this criteria. Interaction is less accurate with touch-screens or motion detection so we shouldn't rely on any degree of accuracy for the user to select objects.

2) Use virtual buttons for different types of selection. There is no equivalent to the difference between an mouse drag and a mouse hover on a touch screen or a motion sencing device so it is better to use virtual buttons and the toolbox metaphore to allow the user to user to choose the different types of selection normally available from an information visualisation interface.

3) Avoid keyboard selection and shortcuts. Touchscreen virtual keyboards are difficult to use and cannot be operated while the cursor is moving. Shortcuts can be better emulated by virtual buttons and incorporated into a tool-box component or gestures on large displays. In the future it might be possible to enter text using speech recognition software incorporated into the system operating system.

4) Move labels away from the cursor so they are not obsqured by the user's finger on touch screen devices.

5) Keep text and selection targets a constant device-specific size and scale other elements to fit these constraints. Text should be readable and the user should be able to make selections using appropriately sized selection targets.

6) Don't display too much data on the screen at the same time. Smaller displays can rely on animation to display more of the data over time and use animation to smooth the transition between views.

7) Avoid sweeping gestures on large touchscreens and motion sencing displays. These can cause fatigue.

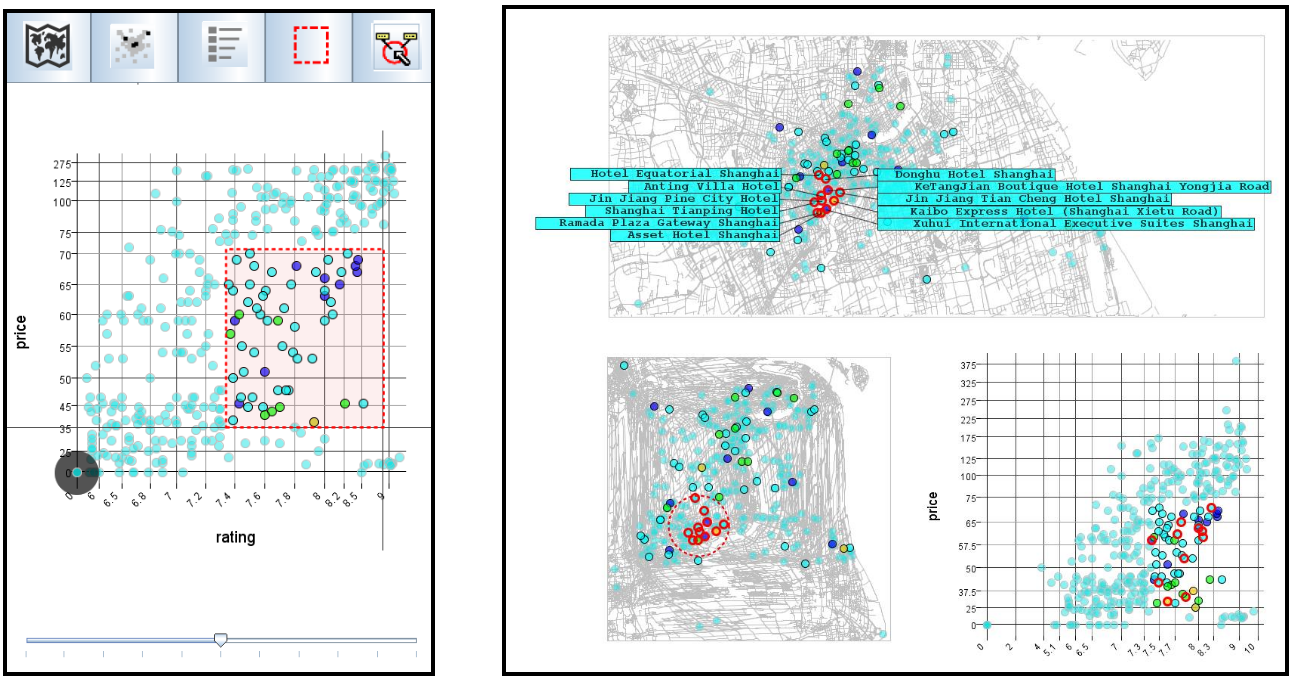

Fig. 1. The HotelFinder Interface for Multi-Device Co-located Synchronous Collaboration. Mobile touch-screen display (left) with coordinated large display with motion detection control (right).

8) Use larger displays to show the data and modile devices for control. Since it's easier to select items from the smaller display and esier to see items on the larger display.

V. PROTOTYPE APPLICATION

Applying these design guidelines to the *HotelFinder* case study allowed us to develop the prototype application shown in figure one.

This shows the mobile display interface with the scatterplot view selected. Buttons on the left hand side allow the user to choose between excentric labelling and box selection, or move from the scatterplot view to the details view or map view. The box selection is being used to select some Hotels with a reasonably low room fee and a higher user rating. When hotels are selected in either the scatter-plot or the map view other hotels are greyed out in all views so as to focus on the user selection. The slider on the right hand side allows the user to apply distortion based on the distribution of the data. This can be applied to have a better view of areas of the map or scatterplot where groups of hotels are normally too tightly clustered together. As the slider is moved, and the level of distortion changes, the points representing apartments gradually move to their new positions so as not to disorientate the user. Buttons along the bottom of the screen allow the user to first label then select a selection made by another user.

Figure one also shows the large display interface. This includes the scatter-plot, map and detailed map on the same screen at the same time. The union of all user selections are highlighted all the users' labels are shown in all views. This allows all users to label and select different hotels without conflicting with other users. The main map view is not distorted so as to give a better representation of geographical distance. This works well since the display space is normally big enough to distinguish between any hotels forming a tight cluster. The distorted map view can be used for a better view of hotels in the city center where the distance between places is smaller and less significant.

Our mobile display requires more interaction to switch between views and operate the animated distortion effect in order to explore the data. More importantly, it also allows the users to operate the shared large display through the coordination of views. This allows the users to compare and discuss their selections toward their common goal of finding a suitable hotel for their shared holiday. In figure one we can see that a user has selected a group of hotels on the mobile screen according to a set criteria and another user has labelled the hotels on the main display to view more detail. This demonstrates multiple users working together toward a common objective.

Our prototype application is fairly rudimentary with basic map views without any details other than the locations of hotels and five notable landmarks. It did however allow us to test a basic case study with three groups of three students planning a visit to Shanghai. The results of a post session survey are shown in tables 1 and 2. Overall the students were impressed by the user experience of the *HotelFinder* and encouraged to use this type of application in the future. The students felt that the interface would improve the level of collaboration and help them to make more democratic decisions when choosing their holiday accommodation. They also felt that the increased insight provided by an information visualization application on a large screen could allow them to find better or cheaper accommodation than they would using a traditional interface. Inter-user conflict was less of a problem than we previously imagined, as the increased level of communication afforded by the interface encouraged the users to act somewhat more amicably. The users did however raise concerns about privacy and practicality as they felt it may be difficult to connect devices or prevent eavesdropping through unauthorized connections. These are issues we hope to investigate in the near future.

TABLE I. USER ATTITUDES TOWARD THE PROPOSED MULTI-DEVICE MULTI-USER INTERFACE

	Response Score	
	Yes	No
Would you consider using this type of interface in the future	87.0%	13.0%
I feel that this type of interface would;	Average[a]	SD
a. Improve the level of collaboration in the group	3.95	0.385
b. Help us make more democratic decisions	3.80	0.470
c. Help us make a more informed decision	3.97	0.409
d. Help us arrive at a better decision	3.95	0.381
e. Improve the experience of working together	3.91	0.411
f. Give us a better insight into the data	3.89	0.457

[a]Responses are scored as follows: strongly disagree=0, disagree=1, neutral=2, agree=3, strongly agree=4. SD abbreviates standard deviation.

TABLE II. USER CONCERNS RELATED TO THE PROPOSED MULTI-DEVICE MULTI-USER INTERFACE

What are your biggest concerns for this type of system?	Response Score[a]	
	Average	SD
Security	4.106	0.338
Privacy	4.200	0.458
Ease of Connection	4.015	0.433
Functionality	3.853	0.424
Ease of Use	4.166	0.365
Efficiency	4.106	0.338
Learnability	3.901	0.441

[a]Responses are scored as follows: not a concern=0, a slight concern=1, a moderate concern=2, a serious concern=3, a very serious concern=4. SD abbreviates standard deviation.

VI. CONCLUSION

We have developed a set of draft guidelines for the development of information visualization interfaces for co-located synchronous collaboration and demonstrated these guidelines put into practice through the development of a simple application for finding hotel accommodation in Shanghai. While the guidelines where limited to a core set of interaction types, through their application they demonstrate the power of information visualization interfaces when applied in a multi-user multi-device environment. In the near future we hope to develop and extend these guidelines by investigating different types of interaction and different capabilities of networked devices. This work will contribute to the vision of pervasive computing where we not only have 'computers-everywhere' but also powerful applications running on and between networked devices to create a more natural, seamless and fluid computing environment.

VII. ACKNOWLEDGMENTS

This work has been supported by the XJTLU research development fund projects RDF140243 and RDF140322, as well as by the Suzhou Science and Technology Development

Plan under grant SYG201516 and Jiangsu Province National Science Foundation under grant BK20150376.

REFERENCES

[1] Clark, H.H. and S.E. Brennan, Grounding in communication. Perspectives on socially shared cognition, 1991. 13(1991): p. 127-149.

[2] Scott, S.D., et al., "Local Remote" Collaboration: Applying Remote Group AwarenessTechniques to Co-located Settings, in Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing. 2015, ACM: Vancouver, BC, Canada. p. 319-324.

[3] Ens, B., R. Eskicioglu, and P. Irani, Visually Augmented Interfaces for Co-located Mobile Collaboration, in Distributed User Interfaces. 2011, Springer. p. 169-176.

[4] Chung, K.-Y., Recent trends on convergence and ubiquitous computing. Personal and Ubiquitous Computing, 2014. 18(6): p. 1291-1293.

[5] Krumm, J., Ubiquitous advertising: The killer application for the 21st century. IEEE Pervasive Computing, 2011. 10(1): p. 66-73.

[6] Weber, R.H. and R. Weber, Internet of Things. 2010: Springer.

[7] Satyanarayanan, M., Mobile computing: the next decade. ACM SIGMOBILE Mobile Computing and Communications Review, 2011. 15(2): p. 2-10.

[8] Isenberg, P., et al., Co-located collaborative visual analytics around a tabletop display. Visualization and Computer Graphics, IEEE Transactions on, 2012. 18(5): p. 689-702.

[9] Seifert, J., et al., MobiSurf: improving co-located collaboration through integrating mobile devices and interactive surfaces, in Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces. 2012, ACM: Cambridge, Massachusetts, USA. p. 51-60.

[10] Huang, X., et al., SecIoT: A Security Framework for the Internet of Things. accepted to appear in Security and Communication Networks, 2015. 8(8).

[11] Sun, Y., et al., Advances on data, information, and knowledge in the internet of things. Personal and Ubiquitous Computing, 2014. 18(8): p. 1793-1795.

[12] Yusoff, N.a.M. and S.S. Salim, A systematic review of shared visualisation to achieve common ground. Journal of Visual Languages & Computing, 2015. 28(0): p. 83-99.

[13] Gutwin, C. and S. Greenberg. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. in Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings. IEEE 9th International Workshops on. 2000: IEEE.

[14] Scott, S.D. and S. Carpendale, Theory of tabletop territoriality, in Tabletops-horizontal interactive displays. 2010, Springer. p. 357-385.

[15] Bashir, R.N., et al., Human Computer Interaction (HCI) in Ubiquitous Computing. International Journal of Innovation and Applied Studies, 2014. 9(2): p. 534-540.

[16] Sakamoto, S.G., L.C. de Miranda, and H. Hornung, Home Control via Mobile Devices: State of the Art and HCI Challenges under the Perspective of Diversity, in Universal Access in Human-Computer Interaction. Aging and Assistive Environments. 2014, Springer. p. 501-512.

[17] Kennedy, J.B., K.J. Mitchell, and P.J. Barclay, A framework for information visualisation. SIGMOD Record, 1996. 25(4): p. 30-34.

[18] Gon, T., et al., Overview "vs" detail on mobile devices: a struggle for screen space, in Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers. 2012, British Computer Society: Birmingham, United Kingdom. p. 233-238.

[19] Deng, Y. and K. Chang, Information Visualization and Location-Based Services on Mobile Devices. 2013.

[20] Craig, P., et al., Information Visualization for the Collaborative Analysis of Complex Data, in 1st Conference on Research and Partnership for Development 2013: Oaxaca, Mexico.

[21] Craig, P., et al. Pattern browsing and query adjustment for the exploratory analysis and cooperative visualisation of microarray time-course data. in Proceedings of the 7th international conference on Cooperative design, visualization, and engineering. 2010. Calvia, Mallorca, Spain: Springer-Verlag.

[22] Craig, P. and J. Kennedy. Concept Relationship Editor: A visual interface to support the assertion of synonymy relationships between taxonomic classifications. in Visualization and Data Analysis 2008. 2008. San Jose, CA: Society of Photo-Optical Instrumentation Engineers, Bellingham, WA.

[23] Nacenta, M.A., et al., Individual and group support in tabletop interaction techniques, in Tabletops-horizontal interactive displays. 2010, Springer. p. 303-333.

[24] Nacenta, M.A., et al., There and back again: cross-display object movement in multi-display environments. Human–Computer Interaction, 2009. 24(1-2): p. 170-229.

[25] 25. Xiao, R., et al. Ubiquitous cursor: a comparison of direct and indirect pointing feedback in multi-display environments. in Proceedings of Graphics Interface 2011. 2011: Canadian Human-Computer Communications Society.

[26] Ahlberg, C. and B. Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. in ACM Conference on Human Factors in Software, CHI '94. 1994. Boston, MA, USA: ACM Press.

[27] Craig, P., N.R. Seiler, and A.D.O. Cervantes, Animated Geo-temporal Clusters for Exploratory Search in Event Data Document Collections, in Information Visualisation (IV), 2014 18th International Conference on. 2014, IEEE. p. 157-163.

[28] Craig, P. and N. Roa-Seiler. A Vertical Timeline Visualization for the Exploratory Analysis of Dialogue Data. 2012.

[29] Roberts, J., et al., Visualization Beyond the Desktop-the next big thing. 2014.

[30] Bongshin, L., et al., Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. Visualization and Computer Graphics, IEEE Transactions on, 2012. 18(12): p. 2689-2698.

[31] Roberts, J.C. State of the art: Coordinated & multiple views in exploratory visualization. in Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on. 2007: IEEE.

[32] Hochheiser, H. and B. Shneiderman. Understanding Patterns of User Visits to Web Sites: Interactive Starfield Visualizations of WWW Log Data. in ASIS. 1999. Washington, D.C., USA.

[33] Fekete, J. and C. Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. in Conference on Human Factors in Computing Systems. 1999. Pittsburgh, Pennsylvania, United States: ACM Press New York, NY, USA.

[34] Craig, P., et al., Assessing the Potential of Colaborative Video Games to Improve Education in La Mixteca Region of Mexico, in Third International Conference on Cognitonics. 2013: Slovenia. p. 413-417.

[35] Craig, P., et al., The Role of Embodied Conversational Agents in Collaborative face to face Computer Supported Learning Games, in 26th International Conference on System Research, Informatics & Cybernetics. 2013: Baden Baden, Germany.

[36] Craig, P., et al., Evaluating the Case for Computer Supported Face to Face Collaborative Learning to Suppliment Traditional Primary Learning in the Mexican State of Oaxaca, in INTED. 2014: Valencia, Spain.