

Real-time Scalable Visualization of Network Data on Smartphones

Avin Pattath*
Purdue University

David Ebert*
Purdue University

ABSTRACT

A variety of mobile devices are available today with varied functionalities ranging from basic phone to integrated cameras, touch screens and full-fledged mobile operating systems. The ability to easily program recent versions of mobile devices (like PDAs, smartphones, etc.) make them a very attractive option to run custom applications, especially, those that need to be executed on the go. One such critical application area is in-field analytics, where we should not only provide easy access to information, but also enable information analysis to some extent.

In this project we present a real-time scalable interactive visual analysis system for network data with other correlated information. It leverages the portability of small screen devices, while banking on its increased computational capabilities on par with a PDA. We use network information collected from Purdue's Ross-Ade Stadium during football games combined with associated video data analysis. Our previous version of the system [6] was based on OpenGL ES and hence did not scale very well across devices. The current system primarily targets smartphones of various resolutions with button-based user inputs. It is designed in OpenVG to make it scalable across all screen resolutions and to provide a consistent interface across various devices. Using the overview+detail [8] approach, it provides fast access to summarized information at a glance and allows a user to explore the information space in detail for detection and analysis of anomalies.

Keywords: mobile visualization, network visualization, mobile visual analytics, smartphone

Index Terms: I.3.2 [Computer Graphics]: Graphics Systems—Network Graphics; I.3.8 [Computer Graphics]: Applications—Visual Analytics

1 INTRODUCTION

In-field analytics, until recently, was limited to the availability of a computing device on site or relegated to remote computers which would send information over the network. However, with the emergence of powerful mobile devices, information and assistance in decision making is readily available to first responders.

However, the capabilities of mobile devices are not without their limits. The quantity of available data poses a big challenge due to the limited processing and storage capabilities of mobile devices. Added to that, limitation in screen resolution (typically of the order of 240 x 240) forces us to explore novel ways to display and visualize information using various abstraction techniques. In this project, we visualize network data and correlated information on a smartphone enabling a system analyst to detect known patterns and discover unknown patterns emerging in real-time.

*e-mail: {apattath|ebertd}@purdue.edu

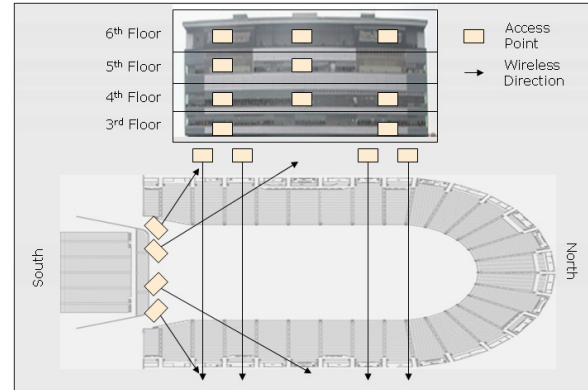


Figure 1: Access point layout in the stadium. It shows 18 of the 20 access points in the visualization, that are typically accessed by users. Eight are installed outside (four near the south score board and the other four on the west side) and the remaining ten are inside the pavilion on the third, fourth, fifth and sixth floors as shown above.

Smartphones are gaining popularity over Pocket PCs today, because of a horde of integrated functionalities (like Wi-Fi, phone, camera, video, etc.). Also, some of the smartphones like Motorola Q, Palm Treo and Sprint PCS PPC 6700 have computing capabilities comparable to that of PDAs. Hence, they seem to be a logical choice of device to develop visual analysis systems where portability is a key factor.

Through this project, our ultimate aim is to investigate the usability of small screen devices for emergency response situations. To this effect, we had developed techniques to visualize network data obtained in real-time from football stadium during home games in our previous work [6]. This particular environment was chosen since network data logs are generally large and sporting events are an excellent test area for such analysis.

While our previous work explored techniques to visualize network data and associated video download logs on small screens, it was not designed to be scalable or consistent. Scalability is very important when a variety of handheld devices are available in the market today. In the present version, we use OpenVG library to make the system scalable to any resolution. To maintain consistency across any device (from handhelds to desktop PCs), we took the overview+detail approach [8] which is easy to navigate using the keys on a smartphone or PDA.

In this paper, we utilize information from Purdue's Ross-Ade Stadium during football games as a testbed. Figure 1 shows the *eStadium* access point (AP) layout in the stadium. Wireless network access information is collected through these APs. We use these network logs, along with the related video and text data, to display various visualizations synchronized with time. Through the combined visualization and analysis of video and text data, we can gain insight into network performance and congestion, crowd analysis, and emergent social behavior. This work provides a proof of concept for mobile sensor analytics that will help to improve network performance.

Our paper is organized as follows: Section 2 discusses previous related work. Section 3 gives a brief description of the *eStadium* testbed. Section 4 discusses in detail, the visualization system we

implemented along with the characteristics of the input data. Section 5 presents our implementation details for Purdue-Indiana game on November 18, 2006. Finally, section 6 discusses some possible extensions for mobile sensor analytics by combining data from various other types of sensors.

2 RELATED WORK

The current project aims to develop visualization techniques to help view time varying network and video data on a smartphone in real time using scalable vector graphics.

Data visualization for analysis purposes has been a fairly researched topic, notably led by the National Visualization and Analytics Center (NVAC) at Pacific Northwest National Laboratory (PNNL) [14, 13]. Several techniques developed at NVAC have influenced the current project. Their temporal analysis methods developed in ThemeRiver [21], for visualizing sequences of word patterns over time has been adapted to help visualize network data over time (for the duration of a game). While ThemeRiver deals with time varying document content, this project aims to analyze time-varying network data. In [7], they also extend the visual analytic platform to mobile devices.

Ubiquity of mobile devices (especially, the more versatile and capable Smartphone) has popularized these devices for many real-time applications. Data deluge in recent applications, has spurred research into adapting data visualization techniques to suit the small screen sizes of the devices. Examples of such work can be found in [9, 12, 17, 11, 19, 8]. Some of the concepts like focus+context visualization can be used in detailed exploration of large datasets. Other examples like the halo technique would be useful when exploring a spatially large dataset like geographical maps as opposed to informatively large dataset (like sensor or network data).

Typically applications developed for mobile devices tend to be mapping tools which are sometimes enhanced with Geographical Positioning System (GPS) [17] or convenience utilities [10, 18]. Usage of mobile devices for emergency situations has been researched by the Measured Response project (headed by SEAS at Purdue) [1]. In this case, they are used for simulation purposes for the first responders to act quickly in case of emergencies. The group at NVAC, PNNL has adapted some of their visual analytics applications to suit mobile devices [7]. Also, we can see an example of usage of PDAs for network data analysis [6]. This project is very similar to the current project with a notable exception that the former has been designed for the PDA screen and developed in OpenGL ES [2]. While OpenGL ES maybe efficient for 3D graphics, it is an overkill for doing just 2D graphics which is mostly the case in information visualization. Hence, the current work will leverage hardware assisted fast 2D rendering library provided by OpenVG [3] which utilizes Scalable Vector Graphics (SVG [4, 20]) to render visualizations that are scalable across any screen size. Also, smartphones have smaller screen sizes compared to PDAs, so many of the visualization metaphors will be adapted accordingly. Since PDAs differ from smartphones in their user input styles (stylus and button input), the current project aims to develop a user interface best suited for button-based inputs.

Temporal visualization combined with network data and geographical visualization can be seen in [15, 22]. Whereas these are pretty complex systems, they can only be operated from a desktop environment or a command center setting. None of them can be operated in-field by a first responder unlike our system which is physically easily portable. The CANVIS system [16] is a network traffic visualization system developed for smartphones, but it is typically not used for visual situational awareness or analytics.

Previous work has been done to provide quality service to game fans during football games through the e-stadium project Living lab developed at Purdue University [5, 23]. The current work aims to complement the e-stadium work by providing the network analysts

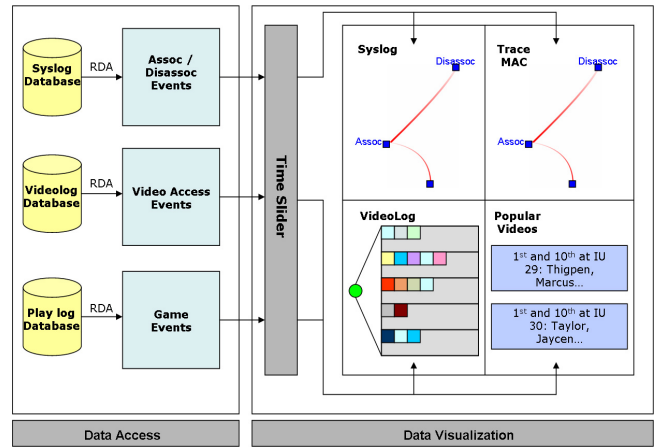


Figure 2: **System overview:** Has two components - **Data Access** and **Data Visualization**. Data Access uses RDA to pull data from SQL database. The visualization component converts the extracted events into visualization components synchronized with the time slider.

with a visualization system to easily monitor network and video traffic accessed through mobile devices.

3 THE *eStadium* TESTBED

eStadium is a long-term, large-scale, collaborative project involving the Center for Wireless Systems & Applications (CWSA), Information Technology at Purdue (ITaP), and Purdue Intercollegiate Athletics [5]. Its main aim is to provide the spectators a great game day experience and in the process solve problems in on-demand delivery over a wireless network of multi-media applications to football fans' PDAs, cell phones, or other portable devices. Further details and an explanation of various terms used in the next few sections can be found in our previous paper [6].

In our project, we use the network and video logs recorded by various APs when fans connect to the network and access *eStadium* applications using mobile devices. The system we develop here is complementary to the *eStadium* project as it helps analysts detect and discover association/disassociation patterns, video access loads on APs, etc.

4 NETWORK AND DATA VISUALIZATION ON MOBILE DEVICES

Mobile devices provide immediate and dynamic access to relevant real-time information in the field. Network analysts may need to make adjustments to load balancing factors, such as wireless AP power settings and preemptive broadcasting of certain types of data, as and when such issues present themselves on site.

With data collected through the *eStadium* testbed, we visualize network, video and text information from Purdue's home games on smartphones. Figure 2 shows our revamped visualization system designed to visualize data using the overview+detail technique. As shown in the figure, the system consists of two different components. To the left is the **Data Access** component, which consists of the databases (SQL Server 2000 on a remote server, SQL Server CE on the windows based mobile device and the "Remote Data Access" (RDA) technique used to pull data onto the mobile device). To the right is the **Data Visualization** component, which shows how we convert data (in the form of events) to various visualization entities synchronized with the time slider.

In the following sub sections, we explain in detail about these two components of our system and their interaction with each other.

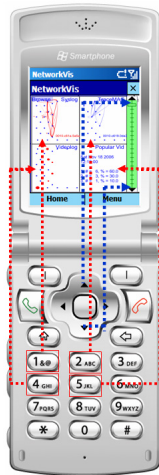


Figure 3: Keys mapped to different regions of the screen on a smartphone. Keys 1,2,4 and 5 are mapped to their respective relative spatial locations on the screen TopLeft, TopRight, BottomLeft and BottomRight. The up and down arrows are used to move backward and forward in time.

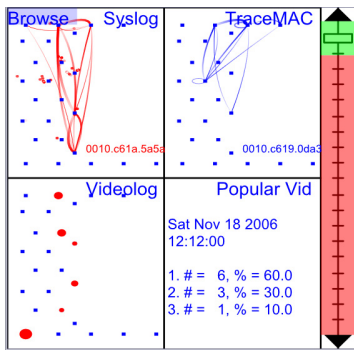


Figure 4: Overview/Context view: Four context regions - TopLeft (Syslog), TopRight (TraceMAC), BottomLeft (Videolog), BottomRight (Popular Video). Time Slider - Green area shows time until when real-time data has been pulled from the database, red area shows the times for which data has not yet been read.

4.1 Data Characteristics

Data for our system comes from several sources, including data acquired from APs (hereafter referred to as “syslog” messages), video messages collected by streaming video servers (as mentioned in the section 3), and a manually-logged “play-by-play” of the sporting event in progress. The messages are logged into a remote SQL database running SQL Server 2000. Mobile devices running Windows Mobile 5.0 are designed to pull data from a remote SQL Server into SQL Server Mobile Edition using the “Remote Data Access” (RDA) technique. RDA is a useful technique when data synchronization is to be done only in one way. Since it does not incorporate data conflict resolution, it cannot be used to keep track of data, but only to replace existing data with new data. Hence, RDA is faster than other means of data synchronization. So, we use RDA to pull data from the server at regular time intervals to incorporate real-time data into our visualization.

We describe in detail, the data abstraction that is used to convert network, video and play logs into event objects suitable for visualization, in our earlier work [6].

4.2 Visualization on Mobile Devices

Mobile visualization system designs have to account for lots of limitations, most notably, screen space and user interaction methods. Moreover, mobile devices come in a variety of screen sizes and resolutions and hence the visualization should be scalable across various devices. While all visualization should display relevant data, the limited display resolution of mobile devices makes the appropriate visual representation very crucial. We have to avoid cluttering the screen, but at the same time display all the essential information in an aesthetically pleasing manner. Hence we need to first extract relevant data at an appropriate level of abstraction, and then match the data to an appropriate visual representation. In the case of network data during a football game, we must also take into account the time variation, and provide synchronized visual display. The following sections describe our approach to solving these problems in our system.

4.2.1 Information Selection

The system handles several different data modes including syslog, video and text. In this case, filtered data not only must convey essential information on its own, but the various modes have to be correlated in order to convey meaningful information. Here we describe how we filter information for the individual data modes.

Syslog Data: The syslog data contains association and disassociation events of each PDA device in the stadium. For visualization purposes, we represent an “event” by a set of disassociation and association activities. By representing this event, we can see the network connection patterns of individual PDAs over time and identify any anomalous behavior. However, for a given time interval, there are numerous such events. Displaying all these events would clutter the small screen and wouldn’t add any useful information. So, we combine the events between a pair of APs in a particular direction and represent them with a single curve whose thickness is proportional to the number of such events. We also appropriately display events from only the most active (in terms of number of such events), or the top few active devices in each time interval depending on the level of detail. This can help in tracking a particular device’s pattern over time. We also display the devices actively associated with each AP. This can be computed by tracking the AP with which a device has an association message without a corresponding disassociation message during a time interval.

Video Data: Video logs from the *eStadium* test bed contain information about how a client perceives videos during a game. Again, displaying all the minute statistics might not be of any practical use since it can crowd the screen. An analyst might be interested in looking at the request patterns of videos from various sections of the stadium. We visualize the number of video accesses from each AP for each different video. Here also we provide different levels of display detail. The system also gives a warning when any AP reaches its maximum video streaming load.

Game Data: The *eStadium* group keeps track of play-by-play events with links to video replays for important events in the game. The information covered in the game data describes significant happenings during the game (for example, touchdowns or field goals). Events are logged by quarters, and events with particular significance are marked with a game-clock time-stamp. We display the text statistics and color code actions (touchdowns, field goals, etc.), from this game log. Also, based on user accesses, we determine and display the three most popular videos in each time interval.

4.2.2 Information Visualization

Our visualization system closely follows the AppLens technique introduced by Karlson in [8]. Their technique was primarily applied to arrange various utility programs on a smartphone for easy interaction while preserving context. While this technique had three modes of views - overview, context and detail, we have only two

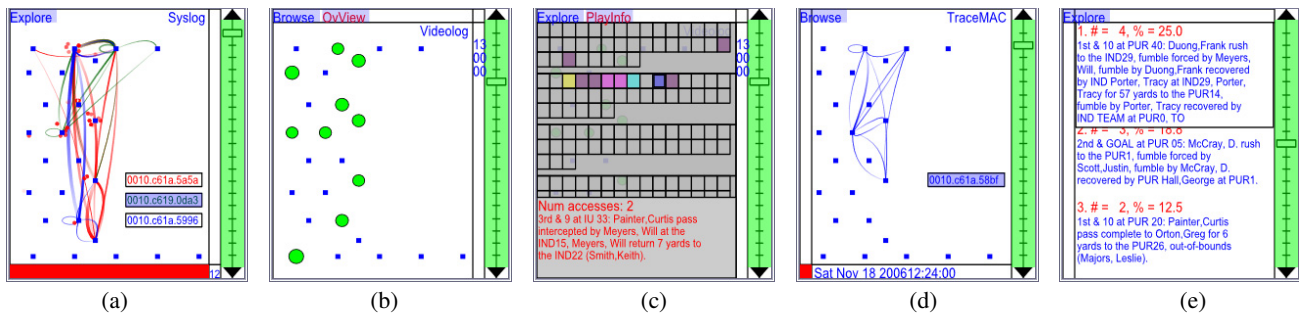


Figure 5: Detailed views of the four different regions. Other regions are compressed and a summarized information, if relevant, is shown. (a)**Syslog data** - top three active devices, with MAC addresses (b)**Videolog data** - Downloaded video statistics (c)**Detailed video statistics** for each video downloaded. Text description of corresponding video is also shown. (d)**TraceMAC** - Trace of device shown in the screen. (e)**Popular Video** - Three most popular videos, with a zoom window for those descriptions that cannot fit in the original screen.

modes - overview/context view (Figure 4) and detail view (Figure 5). In the following sections, we describe the visualization and level of data abstraction in detail, in the two views described above.

Overview/Context View: In the context view, we divide the screen into 5 sections as can be seen in Figure 4. In this view, we don't allow any detailed user interaction, except with the time slider and zooming into the information space for each region of the screen.

The right most section represents the time slider, which is used to step through time intervals. It enables a user to scroll through continuous time intervals and see the visualization corresponding to the time interval between its current and previous positions. The time range for the slider is preset to an hour before and after the football game since the network analysts wanted to examine connection patterns of tailgaters and others who might access the network after the game. The time axis is highlighted with green portion to indicate the time upto which real-time data has been read until then. The rest of the time axis is marked with a red portion to indicate that data has not yet been read for these times.

The rest of the screen space is divided into 4 regions (which I will refer to as top-left (TL), top-right (TR), bottom-left (BL) and bottom-right (BR)). In the TL, TR and BL regions, we show Access Points arranged according to their relative physical locations (shown as blue dots). Four of the APs (the rightmost column) represent the ones that are mounted outside the pavilion, and the row of four APs at the bottom of the screen are the ones inside the south score board. The remaining 12 APs are inside the stadium pavilion on the third, fourth, fifth and sixth floors. The layout of these APs in the stadium can be seen in Figure 1. Two of the APs on the third and fifth floors aren't shown in this figure, because they aren't typically used.

The TL region displays syslog data (Fig 4). At the overview level, this region displays only the association/disassociation patterns of the most active device in each time interval. At the bottom of this region, we display the MAC address of this most active device. It also displays all actively associated devices around the corresponding AP as a red dot (for the current interval) and a grey dot (for the previous time interval).

The BL region displays videolog data (Fig 4). In the overview level, we display only the denied videos from each AP. This is shown as a red circle whose area is proportional to the number of such videos. In this overview mode, user cannot interact with this region.

The TR region (Fig 4) displays association/disassociation movements of a selected device in the stadium. By default, no device is selected and this screen just displays the AP map. The text at the bottom right of this region displays the MAC address of the device being traced currently, and by default, it is set to "No Selection". When a user selects a particular MAC address to track, either from the TL region or this region, only the movements of that device are

shown for all time intervals.

The BR region, in the overview mode, (Fig 4) shows the current time stamp and a list of statistics about the three most popular videos downloaded and viewed by fans in that time interval.

Detail View: The detail view screen is displayed when a user zooms into the information space from the overview screen. This can be done by using spatially intuitive mapped keys (in case of a smartphone) as shown in Figure 3. The zooming process is shown as a smooth animation (in order not to spatially disorient the user), which displays detailed information for the zoomed region and a brief meaningful summary for the compressed regions, if possible. For user interaction in this view, we have defined two modes: "**Browse mode**" and "**Exploration mode**". In the former mode, a user can step through time intervals to update the visualization while being in the detail view. The latter mode can be used for detailed interaction with the currently zoomed region. A user can toggle switch between these modes using the button '3' on the keypad. The currently selected mode is displayed on the top left of the screen as can be seen in the Figure 5(a),(b),(c),(d),(e).

When a user zooms into the TL region (i.e. syslog), he is presented with association/disassociation events of the top three most active devices, each in a different color (red, green and blue in order of decreasing activity of the device) as seen in Figure 5(a). The increasing opacity of the curves indicates direction of a device's disassociation point to association point. Thickness of the curves is proportional to the number of movements of a device between two APs. MAC addresses of the three devices are shown at the bottom right region, in their respective colors. At the same time, the BL region is compressed into a rectangular area, which indicates when any videos are denied, in red color. The BR region would display the minutes section of the timestamp (since this region is the most compressed). Users can select one of the three displayed MAC addresses after entering the "Exploration" mode. This will be added to the TR region for tracking.

The detail view for BL region shows the number of videos accessed from each AP as a green circle around the corresponding AP with its area proportional to the number of videos (Figure 5(b)). It also overlays the information about denied videos as a red circle, if any videos are denied. In this view, in the "Exploration" mode, in a user can cycle through the Access Points on the screen using the arrow/scroll keys on smartphone/PDA. The current AP will be highlighted as the user scrolls through. Any AP associated with video downloads can be selected to view an exploded view of individual video download statistics as shown in 5(c). In this exploded view, a user can scroll through videos (shown as individual boxes) using the arrow keys and select any of them to view the corresponding play description. The color coding is the same as in our previous work. The BR region, is compressed to show just the current time on the time slider.

The TR region (Figure 5(d)) shows the associa-

tion/disassociation events of the device selected for tracking. By default, no device is selected. A device can be selected for tracking from either the scrollable list in this region or from the three devices displayed in TL region. A user can enter the "Exploration" mode to scroll through and select from the list of devices that have logged some syslog events until that time. Here also the compressed regions of the videolog (BL) indicates when videos are denied in red color and the popular video (BR) indicates the time stamp.

The Bottom Right (BR) region in detail view (Figure 5(e)) shows text descriptions of the most popular play videos apart from the statistics shown in the overview mode of this region. Users can scroll through these and view the entire play text, if they have been cut short due to lack of space. Again presence of denied videos are indicated in the compressed BL region.

Real-time data can be pulled at any time using the key 'Enter' at any time. Once the data is pulled, the visualization is updated to reflect the updated data. At each pull, data is read from all the different tables incrementally and appended to the text files residing on the mobile devices. These files are then read incrementally each time, to update the application's data structures.

5 IMPLEMENTATION AND RESULTS

We implemented the visualization techniques mentioned in the previous sections on a variety of environments (Dell Axim x51v, Sprint PCS PPC 6700, Pocket PC 2003/Smartphone Emulator, Win32 simulation). It is scalable across various devices and resolutions (from PCs to smartphones). The only requirement is that it should be running Windows based operating system. The user interaction is suited for keyboard/keypad to support a variety of devices. Stylus/mouse input is not yet supported because of the scalable design.

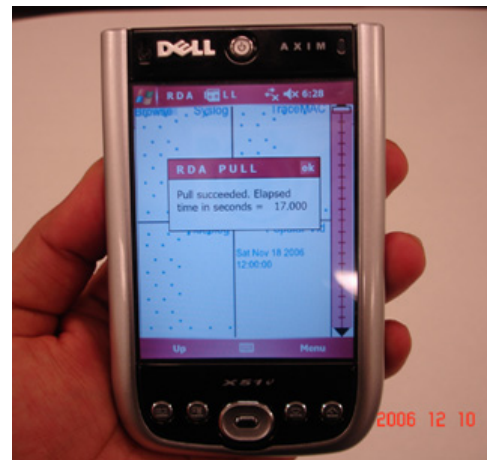
Vector graphics was implemented using OpenVG library to leverage its hardware assisted fast 2D rendering capability [3]. The program was developed using C++ and Visual Studio 2005 IDE. Testing was done on its integrated emulators, and later ported to actual devices mentioned previously.

The data for visualization was collected from the recently concluded Purdue-Indiana game on November 18, 2006. The data is logged into a remote machine running SQL server 2000 database. The data is pulled in real-time through the mobile device (which has to have SQL Mobile installed on it), using a technique called "Remote Data Access (RDA)". We found that RDA was not very fast because it typically took around four to five seconds to pull log data for a period of half an hour. If we pull data for a lesser time interval (like four or five minutes), then, it takes around a second to pull data, which is not very bad for a mobile device. The visualization result screenshots can be seen in Figure 5. These screen shots were captured from a simulated Win32 environment, so that they are clearer to see compared to screenshots of a smartphone screen. Photos of our system running on a PDA (Dell Axim X51v) can be seen in 6. The PDA was set up with the same settings as the Sprint PCS PPC 6700 smartphone, so that the application can run on both these devices.

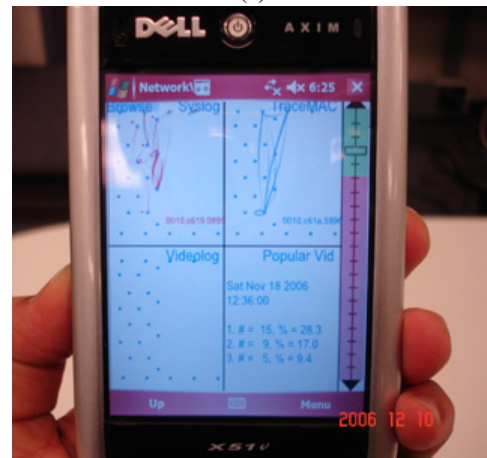
Using the visualization, we are able to do a simultaneous observation of which were the APs with heavy video requests, which individual videos among them had the highest hits, number of hits and what was the related game event. All these observations can be made while preserving the context by looking at the other screens (both in overview and detail views).

6 CONCLUSION AND FUTURE WORK

We have shown in this paper how we can develop an interactive visualization system for analysis of network data in real-time that is scalable across various displays and resolutions. Our system displays time-varying multi-modal data in a synchronized fashion. It



(a)



(b)

Figure 6: Photos captured of the system running on Dell Axim X51v PDA which has the exact environment settings as Sprint PCS PPC 6700 Smartphone.

helps an analyst process data simultaneously without losing context while switching between visualization of different modes of data. Moreover, since it's developed for a mobile device, a user can use it in real time on a football stadium or anywhere else in field for immediate analysis and respond if necessary.

For example, we discovered that some APs were totally free of any association/disassociation, while during certain time intervals there was marked activity between a particular set of APs. Observations such as these will be of use to our network analysts in determining usage patterns and parameter settings of APs.

As future work, we are trying to integrate temperature, acoustic, and video sensor data that will allow us to detect several characteristics that are currently unavailable. Using such data, we can provide estimates on where crowds tend to populate in the stadium and provide location-specific services based on crowd density. Also the integration of emergency response data based on sensor information will be added. By extending the device tracking ability from this system to the crowd, we could provide crowd guidance information to emergency response personnel, such as for optimal exit paths.

ACKNOWLEDGEMENTS

I wish to thank the reviewers for their suggestions and helpful comments. I also thank the National Visualization and Analytics Center (NVAC) for their support. This work has been supported by the

REFERENCES

- [1] [Synthetic Environment for Analysis and Simulation] <http://www.mgmt.purdue.edu/centers/perc/html/>.
- [2] [OpenGLES Overview] <http://www.khronos.org/opengles>.
- [3] [OpenVG Overview: Vector Graphics Acceleration] <http://www.khronos.org/openvg>.
- [4] [Mobile SVG Profiles: SVG Basic and SVG Tiny] <http://www.w3.org/TR/SVGMobile>.
- [5] [e-Stadium - the Wireless Football Experience] <http://estadium.purdue.edu/>.
- [6] Pattath A., Bue B., Jang Y., Ebert D., Zhong X., Ault A., and Coyle E. Interactive visualization and analysis of network and sensor data on mobile devices. In *VAST 2006: IEEE Symposium on Visual Analytics Science and Technology*. IEEE Computer Society, 2006.
- [7] Sanfilippo A., May R., Danielson G., Baddeley R., Riensche R., Kirby S., Collins S., Thornton S., Washington K., Schrage M., Randwyk J.V., Borchers R., and Gatchell D. An adaptive visual analytics platform for mobile devices. In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 74. IEEE Computer Society, 2005.
- [8] Karlson A.K., Bederson B.B., and SanGiovanni J. Applens and launchtile: Two designs for one-handed thumb use on small devices. In *CHI 05*. ACM Press, April 2005.
- [9] Karstens B., Kreuseler M., and Schumann H. Visualization of complex structures on mobile handhelds. In *Proceedings of International Workshop on Mobile Computing*, 2003.
- [10] Bederson B.B., Clamage A., Czerwinski M.P., and Robertson G.G. A fisheye calendar interface for pdas: providing overviews for small displays. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 618–619. ACM Press, 2003.
- [11] Victor C. Information visualization on handheld devices.
- [12] Robbins D.C., Cutrell E., Sarin R., and Horvitz E. Zonezoom: map navigation for smartphones with recursive view segmentation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 231–234. ACM Press, 2004.
- [13] Thomas J. Visual Analytics Challenges and Future Research Areas. In *Workshop on Visual Analytics*.
- [14] Thomas J. and Cook K., editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.
- [15] Goodall J.R., Lutters W.G., Rheingans P., and Komlodi A. Preserving the big picture: Visual network traffic analysis with tmv. In *InfoVIS '05: Proceedings of the 2005 IEEE Visualization for Computer Security*. IEEE Computer Society, 2005.
- [16] Mitchell K., Race N.J.P., and Clarke M. Canvis: Context aware network visualization using smartphones. In *ACM International Conference Proceeding Series; Proceedings of the 7th international conference on Human computer interaction with mobile devices*, volume 111, page 175–182, 2005.
- [17] Chittaro L. Visualizing information on mobile devices. *IEEE Computer*, 39(3).
- [18] Masoodian M. and Budd D. Visualization of travel itinerary information on pdas. In *CRPIT '04: Proceedings of the fifth conference on Australasian user interface*, pages 65–71. Australian Computer Society, Inc., 2004.
- [19] Baudisch P. Halo: A technique for visualizing off-screen locations. In *Proceedings of CHI 2003, Fort Lauderdale, FL*, pages 481–488.
- [20] Liu P. Scalable vector graphics. *IEEE Multimedia*.
- [21] Havre S., Hetzler B., and Nowell L. Themeriver: Visualizing thematic changes over time. In *Proceedings of the 2000 IEEE Symposium on Information Visualization (INFOVIS00)*, page 115. IEEE Computer Society, 2000.
- [22] Kapler T. and Wright W. Geotime information visualization. In *IEEE Symposium on Information Visualization, 2004*. IEEE Computer Society, 2004.
- [23] Zhong X., Chan H., Rogers T.J., Rosenberg C.P., and Coyle E.J. eStadium - the "Living Lab". In *Proceedings of Infocom 2006*, 2006.