ScreenSquared: Improving Media Sharing with Smartphones

Jacob Cole, Katharine Daly, Jesika Haria, Prabhav Jain
Massachusetts Institute of Technology
{jcole, dalyk, jesika, jainp}@mit.edu

ABSTRACT

Among the wide spectrum of today's mobile devices, there exists inherent growing tension between screen size, versatility, and portability. When the iPad was released in 2010, it was said that it looked like a big iPhone, and the iPad is still widely considered the middle ground between the laptop and the phone. The split mode of our application, ScreenSquared, in essence aspires to create an "impromptu iPad," enabling users to share web content across two smartphones by simply placing them side-by-side. In contrast, mirror mode allows two users to each simultaneously view the same content on their own device. In both modes, navigation is possible from either device, and any user actions will be replicated as necessary on the neighboring device. We conducted contextual inquiry studies and semi-structured interviews with 11 users who expressed some concerns about the connection process and the usefulness of various modes (both the ones we implemented plus others they proposed). In general, though, our testers were excited about the concept and felt it was an improvement upon methods of sharing content involving a single smartphone.

INTRODUCTION

The screen size of an electronic device can be a defining feature, as it often dictates how the device will be used. For example, a laptop's large screen size allows a large variety of potential uses but at the same time limits portability. On the other hand, the small screen size of a smartphone makes the device more conducive to carrying around yet minimizes its range of practical uses. In general, screen size is directly proportional to device versatility and inversely proportional to device portability, and it appears that each combination of versatility and portability has a corresponding screen size that would be ideal.

A quick analysis of the current market for electronic devices shows an almost continuous span of screen sizes, with each device attempting to satisfy some different combination of consumer needs. Many consumers, however, don't have sufficient reason or resources to purchase multiple devices, which, barring consideration of convenience, more or less accomplish the same tasks. Certain use cases are perhaps even best accomplished not with any single existing device, but rather with a combination of multiple devices. For example, consider the following common situation: several people are seated together at a dining hall table and someone wants to share an amusing YouTube video. The video is loaded on a single smartphone, and suddenly three or four people must simultaneously hunch over a roughly 4" screen, each struggling to find a suitable viewing angle. The solution to this problem, we feel, is not to design a new portable device designed for multi-person viewing or to expect people to carry a larger existing device with them at all times, but to simply utilize what so many people carry with them nearly all the time - smartphones.

Smartphones have become ubiquitous in the United States, with the number of smartphone users surpassing 100 million in January 2012, according to a study by comScore [4]. Furthermore, Nielsen has reported that 66% of Americans in the 25-34 age range use smartphones [12]. Thus, it would be reasonable to suspect that in the prior dining hall example or any multi-person smartphone viewing instance, more smartphones are immediately available than just the one being used.

ScreenSquared aims to improve social interactions involving smartphones by allowing two smartphones to either split or mirror web content. To establish a connection, users first open the application and then bump their phones together. In split mode, the phones are placed side-by-side and webpages are expanded over the two screens, such that each shows half of the entire view. Users are able to navigate the web from either device (though not on both devices simultaneously) with their actions being reciprocated as necessary on the neighboring device. Most standard internet navigation functions are available along with a split/mirror toggle button that switches between the two modes. In mirror mode, each smartphone displays the same webpage or video in the standard view. Again, navigation is possible from either device, and each screen updates to reflect the users' actions. Figure 1 demonstrates how our app might make sharing web content easier.

RELATED WORK

Multi-screen devices and applications have been explored in the academic literature and are also available commercially. The ScreenSquared app attempts to combine some of the successful aspects of these research studies and improve upon other aspects.

Academia

Chen et al. propose a dual-screen e-book reader which seeks to make the experience of using an electronic reader similar to reading from an actual book. The two screens can either display consecutive pages from a single document or pages from two separate documents, and a

Figure 1: Running the ScreenSquared app on two phones allows users to see the web content more easily than when all four users attempt to simultaneously view a single device.

user is able to switch between these two configurations by attaching and detaching the two screens along a hinge joint. To interpret sensor input from the user and coordinate the two displays, a separate laptop acts as a central server [3].

Along similar lines, Hinkley presents a dual-screen tablet computer called the Codex that changes the content of the two displays based on the relative orientation of the two hinge-connected screens. In general, concave postures in which the two screens face inward correspond to individual use, whereas convex postures where the screens face outward are geared towards collaboration. The collaboration postures by default display the same content on both screens, but the user can enter asynchronous mode using the "send page to other screen" icon [6].

A major difference between our work and that done by Chen and Hinkley lies in the separateness and ownership of the screens. Whereas Chen and Hinkley assume that one user owns both screens, which are part of a single device, our ScreenSquared app aims to coordinate the screen real-estate of two phones owned by two different people, leveraging the ubiquity of smartphones to expand screen real estate on the fly without sacrificing portability. Also, rather than using a central server to coordinate the displays, we pass all pertinent information over a channel connecting the two devices.

Lyons introduces an ad-hoc technique for multi-display composition of Tablet PCs and Ultra Mobile PCs (UMPCs) in which content can either be mirrored on each of the devices or spread across the impromptu screen. It is assumed that the users participating in the collaboration each contribute one device, but like Chen's dual-screen e-book reader, a central server is used to coordinate the displays. Also, the screen size of Tablet PCs and UMPCs is significantly larger than that of smartphones, which we investigate. Interestingly, Lyons's field testing indicates that groups of two using a two-device composition rated their experience higher than groups of four using a four-device composition [9].

The Junkyard Jumbotron, developed by Borovoy in the MIT Center for Future Civic Media, allows screens from any devices capable of running a web browser (including smartphones) to be combined into a larger screen [2]. The connection process, however, requires taking a picture of QR codes displayed on each device, and we hoped to make this setup process faster and more seamless in our ScreenSquared application.

Commercial Products

Though the majority of commercially available smartphones have one screen, the Kyocera Echo has not one, but two 3.5" screens. A special pivot joint allows either one or both screens to be shown at a given time, and when in dual-screen mode, the screens can be used either for different applications or to provide a larger view of a single application [13]. Integrating two screens into one device comes at the cost of introducing additional weight, however, as the Echo measures 6.8 ounces [14].

In December 2011, Kolody released Johnny Test: Roller Johnny, the first iOS gaming experience that runs over multiple screens. The game was implemented using Bluetooth and the virtual screen can span across two iPhones and an iPad [7]. We intend to produce a similar experience across two iPhones with our split mode, which is not specific to a single application, but can be used to view a large variety of web content.

BACKGROUND

We conducted several semi-structured interviews with MIT students to better appreciate our area of interest, namely sharing media using smartphones. We talked to four students in total, asking each roughly the same set of 20 questions pertaining to personal smartphone usage patterns, difficulties with smartphone screen size, methods of sharing

media with friends, and experiences with multi-person viewing of a single smartphone screen.

Types of Media and Duration of Sharing
While their responses were generally in line with our understanding of our interest area, we were pleasantly surprised at the relative narrowness of the range of content that they viewed on a single phone in group situations. The only multi-person viewing instances that were mentioned included YouTube videos (this was often the first example mentioned), Facebook pictures, class websites such as Stellar, Wikipedia pages, and Google searches (C1, C2, C3, C4). Based on this information, we decided that our ScreenSquared app need only allow for multi-screen viewing of web page content as we would be able to improve the vast majority of multi-person viewing instances with just this one capability.

One student (C2) told us that the most common time interval for viewing media in a group setting would be a minute to two minutes, depending on how interesting it was. C1 concurred, saying that four minutes would be the maximum duration and that if the activity were to take longer, then she would get her laptop. We interpreted these short durations of group viewing as implying that our app, especially the connection process, needed to be unobtrusive and quick. With this goal in mind, we designed our app with the minimum amount of setup screens and attempted to make the browsing experience as familiar as possible for our users.

Methods of Sharing Media
When asked how they exchanged interesting content with their friends, our interviewees each had a variety of answers. Methods ranged from describing the content verbally (C1) to emailing links (C3) to instant messaging (C2) to posting on friends' Facebook walls (C3, C4) to pulling up the content on a smartphone and viewing it together (C1, C2, C3, C4). One student (C4) noted that she prefers to show friends videos on their phones because she likes to watch the videos with them to see their reactions. C3 felt that forcing a friend to watch content on a smartphone could be annoying if the friend was doing other things, in which case sending a link might be more convenient. Nevertheless, all our interviewees stated that they had been in a group-viewing situation, initiated either by themselves or by a friend, within the last couple of days (C1, C2, C3, C4). This suggests that despite the wide range of content sharing methods they employ, an app improving media sharing on smartphones would be useful for these students, their friends, and the larger young adult/student demographic they represent.

Small Screen Size
The small screen size was cited by all as major inhibitor to successfully viewing content in groups, and a couple students (C1, C2) mentioned that they often have to pass the smartphone around, viewing in pairs, simply because it is not feasible for more people to see the screen. C4

explained that if she is the one sharing content with others, she often stands back and lets her friends look at the screen since she has already seen it and knows roughly what is happening. Our interviewees said that multi-person viewing could either be done while people were seated (C2, C3, C4) or standing (C3), though sitting seemed to be more common. C4 mentioned an instance when she and her friends were sitting at a table at Toscanini's trying to watch a video on a friend's smartphone since no one had brought their laptop with them. Additionally, although we understand that it is easiest to arrange several smartphones on a horizontal surface, we anticipate that holding phones together for a minute or two while standing, especially only two phones, would not be prohibitively difficult either. In either case, expanding the content over multiple screens (split mode) as seen in Figure 2 would likely improve the ability of multiple people to successfully see the content, hopefully rendering passing the phone around the group unnecessary.

C2 and C3 stated that they often independently read textbooks and newspapers on their smartphones despite the small screen size simply because of the portability of a phone. Should they be able to temporarily borrow a device from a friend, then a split mode would enable them to better see the content themselves. C3 felt that borrowing a device wouldn't be an issue as she frequently lends her boyfriend her smartphone and doesn't mind not being involved when someone else is using her phone. Of course, we appreciate that even if someone doesn't mind lending their phone for a short period of time, borrowing someone's smartphone for a substantial length of time without the lender's involvement is probably somewhat difficult to coordinate in an on-the-go situation. Scenarios in which this use case could apply, however, involve borrowing from family members while at home, on outings, or on long airplane, bus, or car trips.

Figure 2: In split mode, web content is displayed over two screens and navigation is possible from either device.

Figure 3: The first screen instructs users to bump their phone with the other phone that will be used to display web content.

SYSTEM DESCRIPTION

The intent of our ScreenSquared application is to provide smartphone users with an easier method to share web content in groups while minimizing necessary changes in user behavior. This goal corresponds to making the connection process between the two phones as quick and intuitive as possible and subsequently allowing the users to navigate as they would on any standard web browser.

User Interface

Though at the time of the paper prototype usability study, we were using a different connection method than the one we ultimately settled on, feedback from our testers indicated that the desired user behavior to initiate a connection needed to be spelled out more clearly on the

opening screen. Rather than using solely images to indicate a bumping action (the user behavior needed to create a connection in the final version of our application), we also include a text description as shown in Figure 3.

Initially, we built the app to exclusively support split mode; once the devices were connected, all web content was divided over the two screens. However, during the midterm presentations, a significant number of people suggested that we also include a mirror mode where both devices could display the same content. Such a mode might be useful in a situation where two people pass in a corridor and want to share web content with each other. Rather than awkwardly stopping in the corridor, they could instead bump their phones together to connect in a matter of seconds, and each person could continue walking to their unique destination while still being able to watch as the other navigated the web. Both users would have the ability to scroll through the page or navigate to different websites, either by clicking on links or entering a new URL.

For our final ScreenSquared app, we included such a mirror mode in addition to our original split mode and added a toggle button to allow users to easily switch between the two. Figure 4 illustrates these two modes as well as the toggle button which is placed at the bottom of the screen in a toolbar along with back, forward, refresh, and stop buttons.

Technical Details

The complexity of the ScreenSquared application lies not in the screen design but rather in coordinating the content displayed on the two screens. The implementation details can be broken down into several specific components including establishing a connection, constructing the

Split Mode   **Mirror Mode**
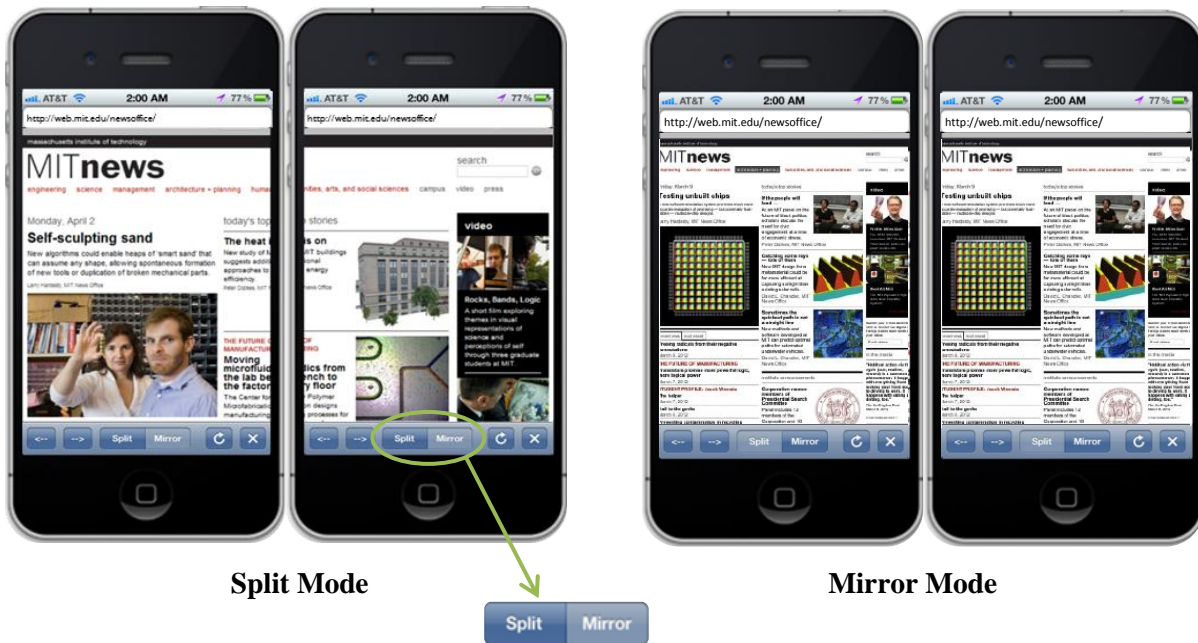
Figure 4: Users can switch between split and mirror modes by using the toggle button in the bottom toolbar.

display, synchronizing events, and handling video.

Establishing a Connection

Before any web content can be shared, the two phones must establish an ad-hoc peer-to-peer connection. We initially planned to accomplish this using the Swÿp API, which would connect devices via Bluetooth as users dragged their finger from one screen to the next and also provide us with a channel over which we could send data. However, the Swÿp library is still in early beta mode and had too many bugs for us to use it effectively. In addition, it is currently intended for communicating sizeable data packets once in a while, a scheme which doesn't necessarily correspond well with the high frequency events that we wish to share between the phones [8].

Thus, we eventually decided to switch to the Bump iOS 3.0 Beta API, which is much more stable and provides most of the same functionality. Bump itself is an app available for Android and iPhones that allows users to transfer contact information and photos to one another simply by bumping their phones together. The Bump API provides an abstract view of this unique connection process and allows developers to access the channel over which information can be sent between devices. Bump operates using any available network connection including WiFi or cellular and uses a combination of accelerometer data and location data from phones to determine which devices should be connected [5]. One tradeoff with using Bump rather than Swÿp is that we can no longer obtain information about the relative placement of the two devices (more specifically, which device is on the left and right). We accepted this handicap however in favor of the better usability and reliability of the Bump API.

In the final version of ScreenSquared, opening our app produces the screen shown previously in Figure 3 which instructs the users to bump their phones together. Once this action is completed satisfactorily, a connection is established and both screens display the Google search page in mirror mode by default.

Constructing the Display

We wanted to make browsing the web in our app as familiar as possible for users, so we emulated Safari's mobile platform. At the simplest layer, the main view of our app is composed of a UIWebView, a UIScrollingView, several UIButtons, a UISegmentedControl for the mode toggle, and a UITextField where users can enter the desired URL.

Without information about the relative location of the two phones, the designation of which device is on the left and which is on the right is done somewhat arbitrarily by determining the lexicographic ordering of the device IDs. When the phones are in mirror mode, webpages are displayed in their standard form, whereas in split mode, each page is zoomed to 200% and offset by an appropriate amount.

Synchronizing Events

Coordinating the two displays required capturing four key events on each phone: scrolling, URL changes, mode changes, and loading videos. To capture these events, we added event listeners to our custom web browser that tracked user's interactions within the browser. As soon as we captured an event, we packaged it into a valid byte-stream format and transmitted it to the neighboring phone. The receiver method on this second phone would be triggered and would artificially execute this same event in order to maintain synchronicity between the two screens.

A difficulty which we had to counter involved infinite loops that would occur when phone A would send a message to phone B, phone B would perform the event, and then phone B would send the same message back to phone A. To solve this issue, we used several flags to keep track of those events which had occurred naturally and those which had been triggered by the other phone. Managing these flags provided extra difficulty, as they had to be reset at the right times and event collisions had to be handled appropriately. This last issue was resolved by attending to the flags according to a strict ordering.

Sharing Video

Video events are the most complicated to handle since iOS automatically launches the native YouTube application whenever a user attempts to play a YouTube video. As we have no control over the native YouTube app, we are unable to programmatically modify the layout or regulate playback of the video as long as the video is played in this manner. Among other approaches, we tried using YouTube's HTML5 API in order to render the video in our own custom player, yet this also resulted in iOS taking control of the playback of the video. As a result, we ultimately ended up writing a PHP script that custom rendered the video and could capture start and stop events on a video playback. Thus, whenever the user attempted to load a YouTube video, we would intercept this request and redirect to our script so that both videos could be started at the same time.

FIELD STUDY

The main focus of our field study was to determine whether our app made sharing web content on smartphones easier or harder by providing two methods (split mode and mirror mode) utilizing two smartphone screens rather than one. We were also interested in understanding how often and under what circumstances testers used each mode.

Methods

It was important for us to directly observe our testers as they interacted with each other and the two devices, for we wanted to ascertain in which situations our app could most comfortably be used. We thus performed a contextual inquiry study followed by a semi-structured interview with 11 students from the MIT fraternity Alpha Delta Phi. Two people (either two testers or one tester with one of us) participated in each contextual inquiry study. For each

iteration of the study, we described the capabilities of the application, gave a quick demonstration of its use, and then asked our tester(s) to use it to share something interesting on the web. After this task was completed satisfactorily, we then interviewed the testers, asking them to compare the ScreenSquared app with current methods of sharing web content on smartphones and to discuss the pros and cons of our app.

Results

Connection Process

All of our testers were able to successfully bump two phones together, although several users expressed more interest in the connection process than we were anticipating. P6 wanted to understand exactly how a "bump" was translated into a connection between two devices, and P1, P4, P8, and P10 were interested in discovering how much distance between the two devices could be tolerated. P4 and P10 decided to run a quick experiment on their own initiative and were able to successfully connect their devices together by simultaneously performing a bumping action at different ends of a hallway.

Users P4 and P6 wanted an alternative to bumping as they felt that the process could be inconvenient, and P6 mentioned the possibility of entering a passcode in order to make a connection with another device. P11 thought that the required setup (pulling out two phones, opening the app on both, and then bumping) might be lengthy enough to prevent him from using the app except in instances where the content he desired to share was lengthy.

Browsing Experience

Based on our generative study, we expected our users to frequently attempt to share content on a small set of websites including YouTube, Facebook, Google, Stellar, and Wikipedia. In general, our 11 testers did follow these expectations, for Google, YouTube, and Reddit were each visited multiple times by different users throughout the field testing. One aspect of the browsing experience that clearly proved time-consuming and frustrating for our users, especially P1, P7, and P11, was entering in new URLs. Our app required users to enter the "http://www." portion of website addresses, but many people naturally left out this prefix, expecting their input to be interpreted correctly. Following the field testing, we made several quick changes based on these observations to our app in order to improve navigation from website to website. These changes are discussed in a later section.

Naturally, creating an impromptu display over several screens does not provide as seamless a view as does a single display, and P2, P6, P7, and P9 commented about the effect of the iPhone border in split mode. P2 and P6 said it felt as if some of the content was being lost along the border even though they realized that it was all actually

being displayed. P7 thought that the border on each of the iPhones was a significant

Several users (P1, P2, P5, P6, and P8) discovered that the multi-person control afforded in mirror mode could lead to "browser wars", in which multiple people are trying to navigate at one time and neither is able to correctly carry out their intended sequence of actions without interruption. For P1, P2, and P5, the browser wars that would occasionally result from natural usage of the app seemed to be more a source of entertainment than a source of frustration. P8, however, suggested designating a dominant controller or allowing only one user to navigate at a given time as he felt that browser wars were undesirable.

The configuration of the devices was also a point of concern for some of our testers. One tester (P7) wanted to rotate the phones into a landscape orientation while another (P9) attempted to swap the left and right phones. Both of these actions are supported in our current ScreenSquared application, so the content displayed on each phone did not change as the phones themselves were physically relocated. Several users also expressed interest in a mechanical add-on that would support the devices during split mode. P1 wanted a stand to hold the two devices together while P5 hoped to hang them side-by-side on a wall rather than being constrained to placing them flat on a table. P10 suggested iPhone cases with magnets that would be capable of at least joining the devices together such that they would not have to be supported independently.

Mirror Mode, Split Mode, and Alternative Modes

In general, our testers seemed to spend roughly an equal amount of time in each of mirror and split modes, as shown in Figure 5. The slight discrepancy between the two is perhaps even explained by the fact that YouTube videos are currently only viewable in our app in mirror mode. In both the contextual inquiry studies and the semi-structured interviews, some users (P5, P6, P9, P10) seemed to prefer one of the modes to the other, while others found both modes to be helpful (P3, P7, P11).

As an example of the former case, P5 stated that he would exclusively use split mode since he did not see any
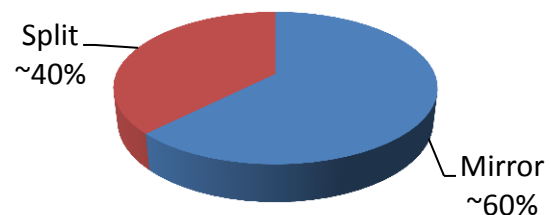
Figure 5: Users found mirror mode slightly more useful than split mode.

difference between emailing a friend a link to content and viewing it simultaneously with the friend in mirror mode. Furthermore, he thought that mirror mode would be inconvenient for sharing something in the same vein as a Wikipedia page, as he would constantly have to ask his friend if he was done reading a particular section before scrolling the page.

P10 echoed this concern with text documents, proposing that there be an option for the devices to be only partially synchronized. In this case, changes in URL would be shared, but scrolling events on one phone would not be duplicated on the neighboring phone. Presupposing the inclusion of his suggestions for a partially synchronized mirror mode, P10 felt that such a variation of mirror mode would be most useful, for he could use it to give impromptu presentations in areas where there were no projectors. P5 suggested a use case of our app along similar lines, wondering if he could have a fifth phone that would act as the master and control four other phones arranged in matrix. Yet another related potential mode was described by P8, who wanted the ability to stop and restart sharing within a short amount of time without having to bump to re-establish a connection.

Comparison to Single Phone Sharing
In the semi-structured interviews, we asked our testers to compare media sharing using two phones running the ScreenSquared app to media sharing using just one phone. More specifically, we wanted to know on a scale of 1 to 5 if using our app made it much more difficult (1) or much easier (5) to share web content with friends. The results of

these individual rankings are displayed in Figure 6.

In general, there were two reasons for the lower rankings. P5, P8, P11 felt that the app was not very seamless, mainly due to the time required to connect the devices and the difficulties associated with navigating to a new webpage by entering a new URL. All three of these users however said that they were fond of the concept behind ScreenSquared and that they would have ranked it better than media sharing using one phone had the implementation been more polished. P5 described our app as a proof of concept and thought that the features it offered were more appropriate as embedded functionalities in operating systems rather than in applications.

Some testers (P5, P7, and P8) also gave our app lower rankings because they felt they had little use for it. P7 said that he rarely encounters cool websites that he wishes to share with people, while P5 claimed that he generally uses his smartphone to view articles that probably wouldn't be of interest to his friends.

Revisions to User Interface
After watching several testers in our field study struggle to enter URLs, we decided to make several quick changes to the user interface to make the browsing experience easier. Firstly, both our initial generative study and our field testing indicated that a small set of websites are common destinations in multi-person viewing instances on smartphones. Thus, providing shortcuts to such common sites as Google, YouTube, Wikipedia, and Reddit seemed an obvious way to reduce the time a user spent trying to navigate to their desired destination. We implemented these shortcuts as four semi-transparent buttons that appear

Improvement Ratings
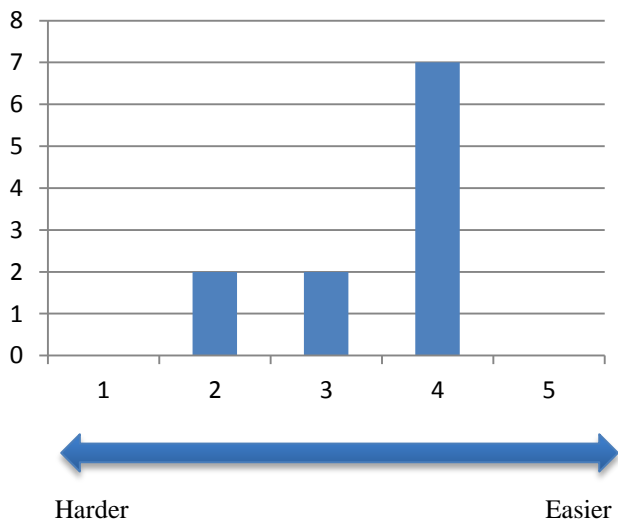
Figure 6: On average, our testers said our app made group sharing of web content on smartphones slightly easier than existing single screen methods.

Figure 7: Shortcuts to commonly visited websites appear when a user clicks the URL bar.

beneath the URL bar when the bar itself is clicked. Figure 7 shows a screenshot immediately after the URL bar has been clicked.

Requiring the entire URL of a website (including the http://www. portion) to be entered was a second cause of a frustrating browsing experience for some of our testers who were inclined to leave it out. To make entering a URL more user-friendly, we thus made entering "http://www" optional; if the user includes it, we make no changes, whereas if the user does not include it, we add it to the beginning of the user's input. We also inserted a delete button on the right hand side of the URL box to prevent users from having to continuously hold down backspace to clear a URL.

DISCUSSION

ScreenSquared represents the emerging trend of smartphones' expansion from individual tools for information lookup into devices to enhance social interaction. There is much concern about the effects of smartphones on culture and the day to day interactions between members of society [11], and there is extensive research into the drug-like neurological effects of smartphone addiction [1]. Thus, as UI visionary Pranav Mistry said in his 2009 TED talk, developments such as this that return users to the real world, that help to "keep us human" instead of "machines sitting in front of machines," are critically important and have major sociological implications as well [10]. By removing a layer of encumbrance in social interaction, ScreenSquared facilitates meaningful connection, proving that these devices needn't be divisive.

One of the deepest original intentions of this project was to reveal the power and convenience of multiscreen applications, inspiring others to follow. Thus, it was meaningful that this original vision of ushering in a new paradigm for accessing mobile devices was echoed in a response from an interview; one person said of ScreenSquared: "This should be an integral part of the OS." This quote, in conjunction with the comment that phones should be able to be "connected with magnets" singularly highlights the crux of our project: the keystone of this family of applications is to facilitate sharing and remove the boundary of the small screen. Its importance is highlighted in the fact that users believe manufacturers must make this a part of their operating system and hardware -- a central feature of the phone. With such a system, we can now share not just screen real estate, but also data, computing power and applications seamlessly across phones. In the past, people were astounded when scientists managed to make computers the size of a room fit on our laps. And yet modern technology, which is able to make this fit in our palms, runs up against a different and much more human limit: screen size. This app provides one of the first practical attempts at solving it. Once integrated, the world

of smartphone sharing and interactions will be rendered boundary less.

FUTURE WORK

Video
Although we were able to implement video sharing for mirror mode using custom PHP scripts, we have not yet accomplished spreading video across multiple screens in split mode. Such functionality is highly desirable since it would allow friends gathered together to comfortably watch movies or movie clips on the go without having to carry an iPad or larger device with them. Our experimentations with video rendering on iOS web browsers did lead us to a few key insights that may help us split video across multiple screens on the future. Foremost, we were intermittently able to, as a proof of concept, parse the obfuscated source of the webpage that the YouTube video resided on and extract an mp4 video. Once we had extracted this video in this format, we were able to use a custom MPMoviePlayer to play the video. We are still hopeful that with this approach (which is officially unsupported by YouTube), we will be able to offer in future versions of the ScreenSquared app a video viewing experience that seamlessly expands across multiple screens. Alternatively, the development of more precise tools for the manipulation of YouTube video feeds may make it more straightforward to create a system for viewing video across multiple screens.

Synchronized Audio
In our generative study, two of our interviewees (C2 and C4) noted that the volume can often be too low when viewing media on a smartphone in a group setting. Being able to perfectly synchronize audio between multiple phones could either complement watching a video in split mode or simply allow a song to be played louder. Currently, a single smartphone is not powerful enough to replace a set of laptop speakers for playing music, but a group of phones could. Moreover, multiple phones speakers together may even be able to construct lower frequency sounds than a single phone at some angles by exploiting the resonance between different phones' sound output (the "beat frequency").

Expansion
Eventually, we would like to support up to n devices operating on various platforms, rather than just two iPhones. Cross-platform compatibility is especially important for our app since we require phone-to-phone interaction, and users should not be restricted from sharing web content with others due to owning devices with differing operating systems. The Bump API already supports interactions between iPhones, iPods, iPads, and Androids, so the main difficulty lies in determining how content should be split amongst the differently sized devices when in split mode.

While arbitrarily selecting the left and right devices is moderately acceptable when there are only two phones, as the number of phones increases, figuring out to which

portion of the impromptu screen each device has been assigned becomes more of a puzzle. Thus at some point it becomes necessary for the users to be able to designate the desired relative positions of the devices. Developing a quick and easy method to set the layout is important, for as P11 related during the field testing, any additional overhead that is part of a setup process reduces the probability that people will use the application.

Different methods for connecting the devices (other than bumping) should also be explored in conjunction with developing a procedure for configuring a layout since several of our field testers did not seem to especially like bumping as a connection method. Preferably, we would return to creating a connection with a finger swipe across the screens to be connected instead of using a bump. While the Swyp API was not yet stable enough to be used in this iteration of our application, we feel that connecting devices with a finger swipe is more intuitive and efficient. This method also allows us to determine the relative positions and orientations of the devices, ensuring that the appropriate phone displays the appropriate content. Moreover, a long swiping gesture allows for the effortless integration of the screens of more than 2 phones – the orientations and positions of an unlimited number of devices can be ascertained through the path the finger traces.

"Browser wars" (users accidentally fighting each other's scroll events) will also become more of an issue as the number of connected devices increases since there is simply a higher probability that more than one person will be touching a screen at once. This is especially true when the devices are being operated in mirror mode and the users cannot see what the others are doing. A potential method for avoiding browser wars could be to only permit a user to control the interface when his or her phone is "holding" a virtual "token". The single token can be picked up by a user only when it is not being actively used by another user, and a "request the token" button could help distribute ownership of the token evenly.

Additional Modes
Feedback from users during our field testing indicated that modes in additional to split and mirror could be useful. Firstly, a broadcast/presentation mode would allow a single phone to control all others. Any user within signal range would be able to join the shared browsing/video session by selecting from a list of detected sessions – no bump would be required. Potentially, browsing sessions could also be broadcast online in real time so people farther away could join in as well. This could be used to give presentations to groups of people in the absence of a projector. In addition, shared audio, if broadcast in real time, could constitute an impromptu PA system. This mode of usage could be particularly relevant to outdoor groups of protesters, as were seen in the Occupy movement and the Arab Spring.

A second potential mode which we call guide mode would be a moderately synchronized version of the current mirror mode. Users would be allowed to scroll pages on their phones independently, while still receiving URL changes. We imagine this mode would be especially useful when text is being shared since each user could read at his or her natural pace while still following each other from webpage to webpage.

CONCLUSION

The ScreenSquared application we have developed is intended to push the boundaries of how smartphones are used in our society as the number of smartphone users in the US continues to rapidly grow. It stands at the forefront of the multi-phone, multi-person web browsing experience as a successful proof of concept, featuring split and mirror modes, video synchronization in mirror mode, and translucent buttons for direct access to frequently visited websites. The "impromptu iPad" approach that our app is based upon is versatile, inexpensive, and useful, and we feel that it is an early example of a new mode of social interaction involving smartphones and other electronic devices in general.

In building this app, we faced non-trivial technical challenges, in that gaining permissions to the sandboxed iOS to synchronize events, especially those pertaining to video, required several workarounds. In our field study, the concept of our application was met with excitement, although there was also a general consensus that more features could be added and more refinements made in order to make the app seamless. In future iterations of the ScreenSquared prototype, we intend to continue improving the interface and integrating various other useful features such as splitting video across screens, synchronizing audio, providing additional modes of use, and allowing n users to simultaneously connect and share content.

REFERENCES
1. Amazon. "Sleeping with Your Smartphone: How to Break the 24/7 Habit and Change the Way You Work [Hardcover]." *Amazon.com: Sleeping with Your Smartphone: How to Break the 24/7 Habit and Change the Way You Work (9781422144046): Leslie A. Perlow: Books*. Amazon, 8 May 2012. Web. 18 May 2012. <http://www.amazon.com/Sleeping-Your-Smartphone-Break-Change/dp/1422144046>.

2. Borovoy, Rick, and Brian Knep. "Junkyard Jumbotron." Junkyard Jumbotron. MIT Center for Future Civic Media. Web. 25 Feb. 2012. <http://jumbotron.media.mit.edu/>.

3. Chen, N., Guimbretiere, F., Dixon, M., Lewis, C., and Agrawala, M. *Navigation Techniques for Dual-Display E-Book Readers.* In *Proc. of CHI 2008.*

4. ComScore. *ComScore Reports January 2012 U.S. Mobile Subscriber Market Share. ComScore.* 6 Mar. 2012. Web. 12 May 2012.

<http://www.comscore.com/Press_Events/Press_Releas es/2012/3/comScore_Reports_January_2012_U.S._Mob ile_Subscriber_Market_Share>.

5. Greany, Thomas. "Bump-api-ios." *Github*. 26 Apr. 2012. Web. 17 May 2012. <https://github.com/bumptech/bump-api-ios>.

6. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, Francois., and Balakrishnan, R. *Codex: A Dual Screen Tablet Computer*. In *Proc. of CHI 2009*.

7. Hutchings, Emma. "Two iPhone Screens Become One In New Mobile Game." PSFK, 30 Jan. 2012. Web. 25 Feb. 2012. <http://www.psfk.com/2012/01/two-in-one-iphone-screen-game.html#ixzz1nSwAqTwr>.

8. Linder, Natan, and Alexander List. "Swyp." *Fluid Interfaces Group*. MIT Media Lab, 2010. Web. 17 May 2012. <http://fluid.media.mit.edu/people/natan/current/swyp.ht ml>.

9. Lyons, K., Pering, T., Rosario, B., Sud, S., and Want, R. *Multi-Display Composition: Supporting Display Sharing for Collocated Mobile Devices*. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part 1,* Springer-Verlag, 2009.

10. Mistry, Pranav. "Pranav Mistry: The Thrilling Potential of SixthSense Technology." *TED: Ideas worth Spreading*. TED, Nov. 2009. Web. 18 May 2012. <http://www.ted.com/talks/pranav_mistry_the_thrilling_ potential_of_sixthsense_technology.html>.

11. National Public Radio. "Separation Anxiety? Taking Cell Phones From Teens." *NPR.org*. National Public Radio, 19 Mar. 2011. Web. 18 May 2012. <http://www.npr.org/2011/03/19/134682348/Separation -Anxiety-Taking-Cell-Phones-From-Teens>.

12. Nielsen. Survey: *New U.S. Smartphone Growth by Age and Income. NielsenWire.* Nielsen, 20 Feb. 2012. Web. 12 May 2012. <http://www.ctia.org/media/press/body.cfm/prid/2133>.

13. Patterson, Ben. "Dual-display Echo Smartphone: Double the Touchscreen, Double the Fun?"*Yahoo News*. 7 Feb. 2011. Web. 12 May 2012. <http://news.yahoo.com/blogs/technology-blog/dual-display-echo-smartphone-double-touchscreen-double-fun-20110207-160516-846.html>.

14. Rodriguez, Armando. "Kyocera Echo: The Phone That Wants to Be Your Tablet." *PCWorld*. 13 Apr. 2011. Web. 12 May 2012. <http://www.pcworld.com/article/225179/kyocera_echo _the_phone_that_wants_to_be_your_tablet.html>.