Journal of Universal Computer Science, vol. 19, no. 8 (2013), 1140-1157
submitted: 30/11/12, accepted: 29/3/13, appeared: 28/4/13

Pick & Drag & Drop: Augmented Reality for Multiple File Sharing

Valeria Herskovic, Carolina Fuentes
(Pontificia Universidad Católica de Chile, Santiago, Chile
vherskov@ing.puc.cl, cjfuentes@uc.cl)

Richard Ibarra
(Universidad de Chile, Santiago, Chile
richard.ibarra@gmail.com)

Javier Bustos-Jiménez
(Nic Chile Research Labs, Santiago, Chile
Universidad de Chile, Santiago, Chile
jbustos@niclabs.cl)

Abstract: Many users nowadays work with multiple heterogeneous computing devices, such as desktop computers, tablets, laptops and mobile phones. Transferring files between devices is cumbersome and usually done through the internet (e.g. email or cloud computing services) or physical devices such as flash drives. Some solutions for this problem have been proposed, however, they do not allow efficient and easy transferring of several files over medium distances. To facilitate file transfer between different devices, we implemented an augmented-reality based smartphone application that uses “pick-and-drop” and “drag-and-drop” mechanisms to transfer files between devices. This paper presents the implementation and evaluation of the interaction technique, called *pick and drag and drop*. This technique has two central contributions: it allows transferring several files at once, and it allows file transfer over greater distances than existing alternatives. The technique was tested in two laboratory experiments with promising results: time taken to transfer files was similar to other options, and 78% of users declared preferring PDD to alternative methods.

Key Words: Augmented reality, Data sharing, Mobile, Multiple devices, Pick and drop

Category: H.5.1. Information Interfaces and Presentation: Multimedia Information Systems: Artificial, augmented and virtual realities.

1 Introduction

Many users have several computing devices in their work area, a combination of e.g. desktop, laptop, tablet and mobile devices. Each device is usually clearly distinct from the rest, having e.g. different sizes, languages, and operating systems. However, there is a lack of direct manipulation interfaces to work across the boundaries of multiple devices [Rekimoto, 2000]. For this reason, it is currently not easy to do a simple task such as transfer files from one device to another, even if they belong to the same owner, and many users are still using mechanisms such as email, FTP or cloud services [Chang and Li, 2011] for this task.

Currently, most persons own a mobile phone: mobile phone penetration is over 96% in the U.S. and surpasses 100% in other countries. Mobile phones have become an essential and extremely personal tool. More than 30% of U.S. mobile phone subscribers own *smartphones* [The Nielsen Company, 2011], or mobile phones with considerable computing power. The processing power and hardware capabilities of smartphones, as well as their large user numbers, have driven the development of applications, among them augmented reality. Augmented reality (AR) allows a user to see the real world in combination with virtual objects, usually superimposed on top [Azuma, 1997]. For example, there are currently several AR frameworks and applications on the iTunes App Store and Android Play Market, such as Mixare, Layar and Wikitude.

This paper proposes an AR-based interaction technique as an approach to the problem of transferring files between different devices called *Pick & Drag & Drop (PDD)*. The technique allows a user to view the computer screen through his smartphone, tap on a file through the smartphone screen and carry it to another computer, providing a simple interaction technique that only requires one tap to copy, one gesture to drag, and one tap to paste a file. The contributions of this work are the following: the technique allows multiple file transferring, and it is possible to transfer files over longer periods of time and larger distances than alternatives.

The rest of the paper is organized as follows. First, we present a motivation vignette to illustrate the type of scenario PDD is aimed for. Next, we discuss several approaches that have been proposed recently to solve related problems. Then, we present our proposed interaction technique, and discuss the implementation of our approach. We present a two-stage laboratory experiment, and results from this evaluation. We discuss these results, and then the final section presents conclusions and directions for future work.

2 Motivation Vignette

Picture the following scenario:

Alice works at a university, where she has a desktop computer in which she does most of her work. She also has a laptop computer that is lightweight and therefore used to occasionally work from other locations, such as outside, in meetings, or to teach class. She also owns a personal smartphone. Alice attends a meeting with her colleagues, who are discussing the department budget. She has a powerpoint presentation on her laptop, which she presents to her colleagues. One particular number corresponding to the budget is questioned by the other professors. Alice wants to clarify how she got the number: to do so, she wants *to show her co-workers an excel file with her calculations, a PDF file with the scanned invoices corresponding to the expense report, and a university memo describing the correct way to file expenses. However, these files are in her office in her desktop computer, as she has not anticipated requiring them. To solve this problem, she ponders her options:*

1. ***E-mail or Cloud****: Alice may search her e-mail or cloud folders to try to find some old version of the files. She does not want to include all the attendees in her search, so she may disconnect her laptop from the display and conduct the search or configure her laptop with the extended desktop feature to search privately and then re-configure the display to continue the discussion. However, she knows she only has the latest version of the files she needs on her desktop in her office.*

2. ***Pendrive or Flash Memory****: Alice may return to her office to retrieve the necessary files. She may need to find a flash drive, connect it to her computer, wait for detection, transfer the files, eject the flash drive, and then go back to the meeting, connect the flash drive and open the necessary files. In this option, she interrupts the meeting but has the option of looking for her files in private. However, it may be difficult to find a flash drive if she had not anticipated this need.*

Alice must select the option that is best for her present scenario. Although the exposed options solve the problem, they are cumbersome or represent a violation of Alice's privacy. We propose an interaction technique, PDD, in which Alice would solve the situation in the following way:

Alice picks up her smartphone, which is always with her, and walks back to her office. She quickly finds the files on her desktop, turns on the PDD application on her phone, points her phone at the screen and drags the files she needs to it. She then walks back to the meeting, points her phone at her laptop, and drags the excel and memo to the laptop, resuming her presentation with the needed files. She passes her smartphone to a colleague so he can check, in the meantime, the scanned invoices.

3 Related Work

As far back as 1997, when multi-computer environments were less common than today, Rekimoto identified the problem of transferring files between multiple devices and proposed a pick-and-drop solution for multiple computer environments, based on a pen used to transfer a file between two devices [Rekimoto, 1997]. The pen did not actually carry the files, but rather signaled the involved computers to transfer the files seamlessly through the network. Several other pen-based techniques for "multi-display reaching" have been proposed, e.g. gesture-based copying, and throwing elements between devices [Nacenta et al., 2005]. These techniques, however, require an additional device not belonging to the user's routine to transfer files, they are for single file transfer, and they also do not effectively transfer the files to the intermediary device (i.e., if the process was interrupted at some point the user would not have access to the files).

More recent work takes advantage of the capabilities of current technologies, e.g. cloud computing and smartphone hardware. The most notable and recent examples are summarized below in chronological order.

– *Touch & Connect and Touch & Select* [Seewoonauth et al., 2009] use near-field communication (NFC) to copy a selected file from e.g. a laptop to a mobile phone, which requires short distances between devices.

– Boring et al. [Boring et al., 2010] proposed video-based interaction, especially for distant displays. Their *Touch Projector* allows users to use smartphones to manipulate content from a user's perspective. Specifically in the file copying task, users can select a file on a target display through their smartphone and drag it into another display.

– For cloud computing, *SPARSH* transfers data through touching one device, and then touching another device to paste the copied data, while transferring it through cloud computing services on the background [Mistry et al., 2011]. This application requires all the devices to have touchscreens, and for a user to stay logged into the cloud services as long as s/he wants to transfer files.

– *Deep Shot*, a smartphone application and framework, is used to take on a more complex task: it takes a picture of an application running on a device and allows seamless use on the mobile phone or transferring it to another device, in this way sharing running tasks between devices [Chang and Li, 2011]. Deep Shot is used to transfer application content instead of files, so the content type must be supported in both devices, and the task must be running to allow copying content.

– *PhonePick&Drop* and *PhoneCopy&Paste* [Schmidt et al., 2012] were proposed to transfer data objects from a surface computer to a phone and vice versa, by privately selecting a file on a mobile phone and touching the surface with the phone.

Our proposal, named *Pick & Drag & Drop* (PDD), extends previous work, using an interaction technique that provides two main improvements over previous proposals. First, PDD allows users to copy several files. This is done through a sidebar into which users can drag their selected files. Second, PDD allows users to carry files between locations (it can carry files between different physical and network locations) as in our motivation vignette. This is because in our proposal, the files are actually copied to the intermediary device and the user does not need to physically drag the file since it is kept in the sidebar. Other works, such as Boring et al.[Boring et al., 2010] were proposed for another use scenario, e.g. to manage distant displays, so all devices are required to belong to the same network and be co-located.

4 Augmenting Computers: Pick & Drag & Drop

Mobile phones are pervasive in users' lives, and are very personal, with users who are emotionally and physically attached to them [Srivastava, 2005]. This attachment can be seen, e.g. in work and social meetings, in which frequently, we see each person place their mobile phone on the desk or table to be permanently visible as they are working or socializing.

This metaphor of the phone as an extension of a person is our motivation for developing a pick-and-drop application for smartphones: picking up a file with a mobile phone should be almost as natural as picking up a file with your hand. Therefore, this proposal is in the area of hand-held video see-through augmented reality [van Krevelen and Poelman, 2010]: AR is overlaid on top of a video feed of reality, and the device presenting the AR layer is a hand-held smartphone. The proposed technique, named Pick & Drag & Drop (PDD) allows a user to copy files from one device to another using a smartphone. Next, we describe the particular aspects of our interaction technique and its implementation.

4.1 Interaction Technique

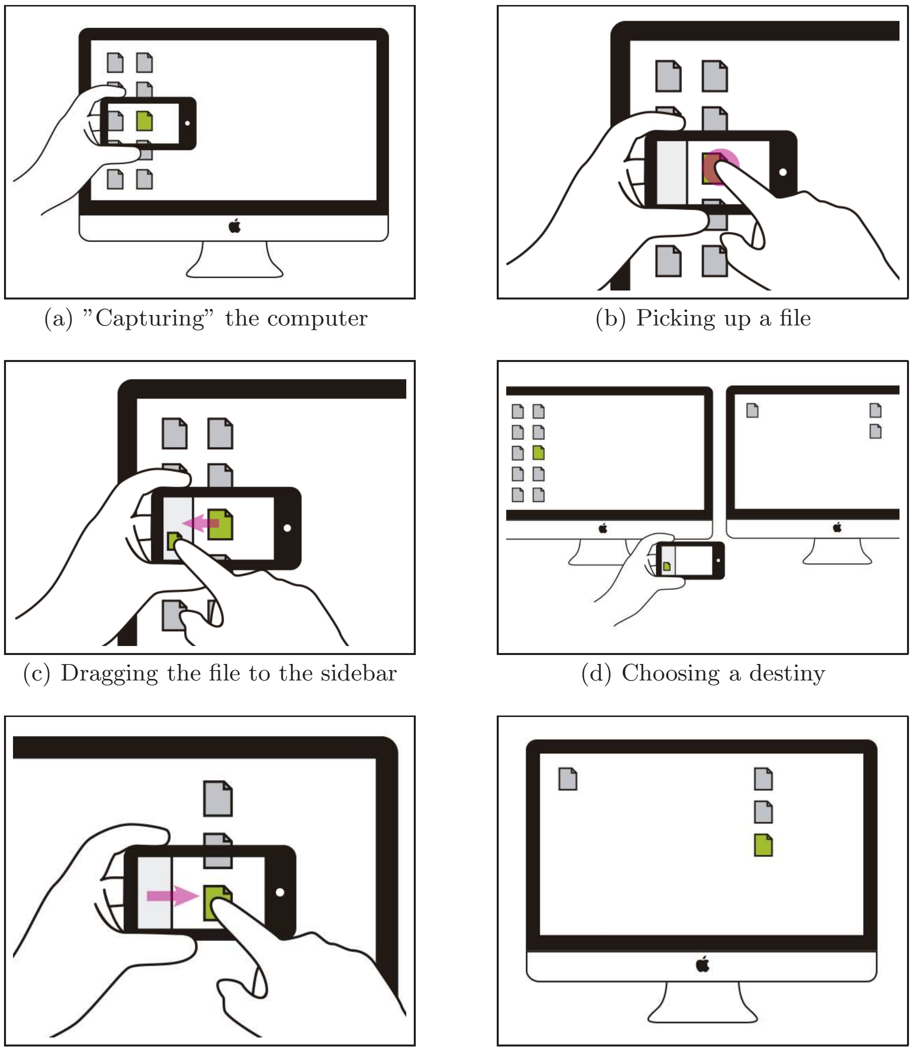
The use of the application is the following: first of all, in the source computer the folder containing the fie to be copied must be open, or the file must be on the computer desktop. Then, the user opens the smartphone application, which looks almost exactly like the phone camera, and points the camera at his computer screen (Fig. 1(a)).

The user selects the file by tapping on it on his phone screen (Fig. 1(b)), and drags it to a sidebar where it will remain (Fig. 1(c)). The user can therefore keep selecting as many files as s/he needs on the computer and "parking" them on the sidebar until it is full (6 files in the current prototype, configurable by decreasing icon size). These files are actually copied to the device, so if the user wants, he can interrupt the process until later.

Afterwards, the user superimposes his phone on the receiving device (Fig. 1(d)). Then, the user can drop all the files it wants to the new location by using his finger (Fig. 1(e)). The files will be transparently copied to the receiver computer, appearing in the available desktop (or Finder folder) (Fig. 1(f)).

It is important to note that the files are copied both to the mobile device and to the receiving device, so files may be dragged over significant distances and into other networks, as long as the mobile and receiving device will belong to the same network. Also, note that this proposal allows copying several files.

The sidebar has a *trashcan* icon on the bottom left, so users can delete files they copied accidentally or will not use again.



(a) "Capturing" the computer

(b) Picking up a file

(c) Dragging the file to the sidebar

(d) Choosing a destiny

(e) Dropping the file

(f) Final status

Figure 1: Use of Pick & Drag & Drop

Chang and Li posit the DeepShot interaction technique is as simple as using a camera on a mobile phone [Chang and Li, 2011]. In the case of PDD, the interaction technique is only marginally more complex: users must point the device towards the computer screen as if taking a picture, but point to a specific file and drag it to the sidebar. To drop a file the interaction technique is the reverse: they must select a file from the sidebar and drag it to the desktop as seen through the mobile phone. The additional actions consist of *dragging and dropping* in both directions (Figs. 1(c) and 1(e)).

4.2 Implementation

We implemented two applications: a mobile client using *Objective C* (for iPhones) and another in *Java* for desktop or laptop servers. Naturally, in the future we plan on implementing the client side of PDD on several other platforms to allow file sharing between different types of devices and operating systems (e.g. Android).

PDD uses *zero configuration networking* techniques for resource discovery, and a simplified screen recognition procedure (based on a predetermined number of possible files and background colors) is used to detect which computer is being watched by the smartphone. This simplified procedure is due to the fact that PDD is a prototype application to test a new interaction technique. Several approaches to more advanced screen detection and OCR are discussed in e.g. [Herbert et al., 2011].

Zero Configuration Networking (a.k.a *zeroconf*) [Guttman, 2001] is a set of techniques for creating and managing a usable IP network automatically and transparently for the users, without the intervention of third parties (manual operator, configuration servers). PDD uses *zeroconf* to inform an iPhone which other devices or computers are also connected to the local network, updating its peer table. It is important to note that since PDD uses Bonjour, it currently only works correctly if both involved devices (mobile client and server) are connected to the same network (but the originating server and the receiving server may be in different networks, as long as they are, at the time of copying files, on the same network as the mobile client). The protocol works "à la Bonjour": first the iPhone sends an exploration message to the network (`HELLO`) to see who else is connected. The *Access Point* forwards the message via the multicast channel to every connected device (Figure 2). Then, all devices who receive the exploration message will send their credentials to the iPhone. The iPhone is responsible for updating its peer table regularly.

From the system's point of view, PDD can be divided in two stages: "Pick & Drag" and "Drag & Drop".

For the "Pick & Drag" stage, the steps are the following:

– each time a file is *picked* for 2 or more seconds, a low-resolution screenshot is taken from the phone camera. OCR is used to (try to) detect the filename

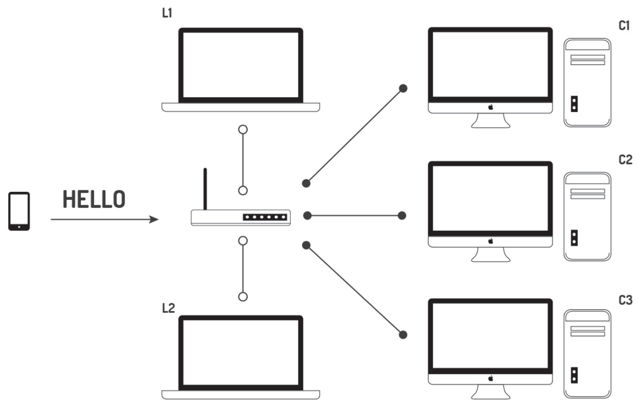


Figure 2: Multicast DNS used as Apple's Bonjour service.

and the screen background color is detected to help identify the computer being watched,

- when the icon is *dragged* to the sidebar, both the detected color and filename are sent from the iPhone to its *peers*, asking "*who am I watching now and has this file?*",
- each computer device will compare the detected color against the prominent color of their Desktop's Wallpaper and, if it matches, they take the filename and check if that file exists at the Desktop (or at an open Finder window).
- then, the computer sends the file to the iPhone.

For "Drag & Drop" stage, the steps are quite similar:

- when the icon is *dragged* from the sidebar and *dropped* to the videoview, a screenshot is taken from the camera,
- the detected color on the screenshot is sent from the iPhone to its *peers*, asking "*who am I watching now?, I have a file for you*",
- all computer devices will compare the color against their own view of Desktop + Finder Windows. If the color matches, the computer sends a message to the iPhone replying "*it's me*".
- the file is sent to that computer.

As it can be observed from the implementation description, currently the computers are detected through their background colors, which requires each participating computer to have a significantly distinct wallpaper (ideally a plain color, or e.g. one with green leaves, another with a picture of the sea). This approach is limited and may be improved and generalized using techniques presented in e.g. [Herbert et al., 2011].

4.3 Technical constraints

PDD was implemented using an iPhone 4 smartphone and Apple MacBook laptop computer. This limits the variability in e.g. resolutions and screen sizes, which is positive for application performance. However, in a real setting these are important factors to consider when transferring files in multiple computer environment. This section summarizes the main aspects which, in our experience developing PDD, must be considered when implementing this type of application. These encompass characteristics relating to the smartphone, the target computers, the environment where the application will be used, and the actual use of it.

- **Smartphone**: Currently, smartphone camera resolutions vary. The iPhone 4, which was used for development, has a 5MP camera, while the iPhone 4S and 5 have 8MP and better quality lenses, performing more favorably in lower lighting conditions. Android phones present even greater variability, so a phone in the lower range may have additional difficulty detecting file names and screen backgrounds.
- **Computer**: Computer screen resolution is another factor which may affect file and computer background detection. Also, screen sizes may vary (it is possible to even consider copying files from projectors [Boring et al., 2010]) and with this, the user must adapt by adjusting the smartphone distance to the computer.
- **Environment**: The environment of use is important. Low lighting, or too much brightness (e.g. if used in direct sunglight) may affect usability.
- **Use**: In most use cases, the user naturally will adapt the distance between phone and computer so that the filename is readable from the smartphone. However, some users may want to copy a file by standing far away (e.g. at a presentation) or too close for the filename and background to be recognizable. Users may also have difficulty with their hands being too shaky to operate the application. Pictures that capture other elements besides the computer screen (edges of the screen, background and foreground elements outside of the computer) may make screen recognition more difficult. Also, computers with screens at odd angles, or smartphones used at odd angles, may also affect detection.

5 User Study

We performed a two-stage user study to evaluate the use of the Pick and Drag and Drop interaction technique. We wanted to evaluate whether this interaction technique is easier to use than existing alternatives, and whether the sidebar helps users transfer several files over a distance more effectively than other alternatives.

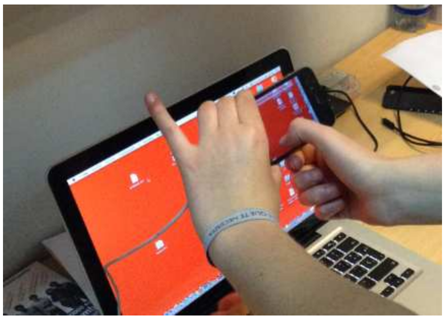
5.1 Task

The experiment was as follows. First, users filled out a questionnaire intended to register demographic data and gauge their experience with computers and file transfer. Then, they were briefed on the use of the application and given a one-page manual with a diagram about PDD use. Participants were then asked to enter the room where the experiment took place and perform the following task: they had to transfer four specific files from computer A (red background) to computer B (blue background). The users were instructed to pick up all four files, dragging each to the sidebar, and then walk over to the other computer and drop them. An evaluator recorded video and took pictures of the process. If the transfer of some files was successful, the evaluator finally demonstrated that the files had effectively been transferred by opening some files in computer B. Users were then asked to fill out an exit questionnaire with their impressions. The second experiment used the same task, but asked users to repeat it using a pendrive, in order to compare time taken to complete the task and user experiences. The users in this case were randomly assigned to do the task first with the pendrive or with PDD.

5.2 Participants and Apparatus

The experiment was conducted in an office, using two laptops: a MacBook Pro and a MacBook Air (both with the server application), and an iPhone 4 smartphone (with the PDD application). Figure 3 shows a user performing the picking up and dropping files during the experiment.

For the first stage, we recruited 20 engineering students, with ages ranging from 18 to 21 (average: 19.9), as volunteers. The students were not compensated. Three of the participants were female. Fourteen of them owned a smartphone, and all of them owned computers and used the internet frequently. The students reported using as file transfer methods: their email (16), external memory such as flash drives and hard drives (15), and cloud computing services such as Dropbox and Google Drive (14). In the second stage, we recruited 12 new engineering and graduate students as volunteers, with ages ranging from 20 to 35 (average: 24.9). Three of them were female, 11 owned a smartphone, all of them used computers and the internet frequently. They reported using their email (11), external memory (10), cloud computing (9), and bluetooth (1) as file transfer mechanisms.



(a) Picking up a file

(b) Dropping the file

Figure 3: User testing Pick & Drag & Drop

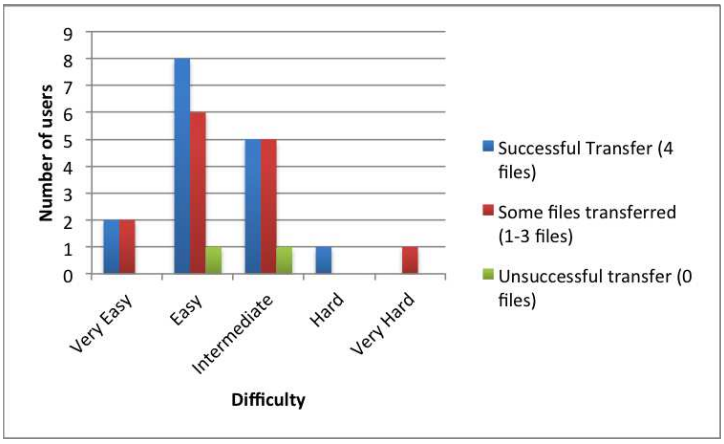


Figure 4: Difficulty of use vs file transfer status

5.3 Results

This section discusses the results from the experiment. We have three types of results: first, we present results from the questionnaires (before and after the experiment). Then, we briefly discuss the comments from users during and after their participation, recorded through video and later transcribed. Finally, the experiment, combined with personal experience transferring files, allowed us to visualize the main items in which time is spent when transferring files, so we propose a theoretical way to compare file transfer mechanisms and apply this to PDD and other alternatives.

5.3.1 Experiment results

Although two users were not able to transfer files and 14 others did not transfer all files, over 78% of users reported they would prefer PDD over other alternatives they typically use, if the application was available and did not present any usability problems (discussed in section 6). Moreover, when asked about the ease of use, more than 59% of users (independent of operation success) believe that PDD is easy or very easy to use (Fig. 4). Section 6 discusses reasons for unsuccessful transfer and ideas for improving success rates.

The experiment also compared transfer times. After using PDD users were asked to estimate the time taken to transfer four files through PDD versus the time taken by their usual preferred transfer method. This self-reported time is not exact, but allows us to understand the perception of users about the efficiency of PDD. It also compares PDD against the best transfer method users could think of. All users perceived PDD to be equal or faster than existing alternatives. On average, PDD was considered 37% faster than other methods.

In the second stage of the experiment, users were asked to transfer the same set of files through PDD and a flash drive. The evaluator timed the users for both setups: the time taken to transfer the files from the first computer to the smartphone or flash drive (t_1), and the time taken to transfer the files from the smartphone or flash drive to the second computer (t_2). For the 12 participants of the second stage of the experiment, the average times are shown in Table 1. We may see that the times for both mechanisms are similar: PDD was faster in the first transfer stage, while the flash drive was faster in the second transfer stage.

Table 1: Time comparison for both file transfer stages

	Flash drive	**PDD**
t_1 *(s)*	24.01	21.83
t_2 *(s)*	25.98	29.49
total	49.98	51.32

5.3.2 User comments

Users also spontaneously commented on the use of the system. Most comments about the application were of two types. One type of comment expressed surprise and enthusiasm when the file transfer worked (e.g. "are files really transferred? it's like magic!", "it's so weird, but it really works"). The other type of comment remarked that they felt PDD was very simple (e.g. "this is so simple it's fool-proof"). Other comments gathered from the videos were of the users thinking aloud about what they were doing (e.g. "oh, are they transferring?", "I'm doing this file by file, is that ok?").

5.3.3 Theoretical results

This section compares, in a theoretical way, the total time taken to transfer files via several methods. This is a proposal based on the authors' experience with each file transfer method, in which we posit that the total time taken to accomplish the transfer of n files is a function of the time of setup, the time to transfer each file, the distance between the source and target computers, and the time to conclude the process.

- *Setup*: Considers setup time such as loading an application, finding a file, setting the distance from the computer, etc.
- *File transfer* : Corresponds to the time to transfer each file, when the file transfer mechanism has already been set up.
- *Conclusion*: Corresponds to the time to setup the transfer to the second computer, e.g. connecting a flash drive, configuring an application to work with another computer.

We assign a time - either high (H), medium (M) or low (L) - to each of these tasks. The time to traverse a distance d is t_d. For example, in the case of transferring four files with a flash drive, the setup time is the time taken to detect the drive (M), then for each file it must be selected and dragged to the drive (L), and when at the target computer, the flash drive must be detected again to enable copying the files (M), and the files must be copied to the second computer (L). Then, Table 2 presents the time taken to transfer n files over a distance d. Table 2 allows us to determine in an approximate and theoretical way the differences between the use of PDD and other methods such as a flash drive and Touch Projector [Boring et al., 2010]. As a result, we can see that PDD is marginally faster than e.g. a flash drive, as it does not need setup time in the target computer, and faster than systems such as Touch Projector (when n>1) where each file has to be carried individually. Another common file transfer mechanism is cloud computing, which is somewhat different to the previous techniques: in case the user has the cloud storage set up in both devices, file transfer is quite easy and transparent, and much faster than the other available methods. However, if the system is not setup, and software must be downloaded, configured, and all files must be synchronized before the needed files are transferred, this technique is slow and cumbersome. This is similar to transferring files through e-mail as well.

Table 2: Transference time for distance d and n files

Method	Setup	File	Conclusion	Total
PDD	M	$2L$	0	$M + 2nL + t_d$
Flash Drive	M	$2L$	M	$2M + 2nL + t_d$
Touch Projector	M	L	0	$M + n(2L + t_d)$
Cloud (already setup)	0	$2L$	0	$2nL + t_d$
Cloud (not setup)	H	$2L$	H	$2H + 2nL + t_d$

6 Discussion

In the questionnaires, users reported using several file transfer methods (e.g. 78% reported using flash drives, 71% reported using cloud computing services, 87% reported using e-mail). The different reported file transfer methods suggest the type of file transfer is chosen according to the particular situation. We posit that PDD can coexist with other file transfer mechanisms and be used in situations when it is the best solution, as in our motivation vignette. PDD is similar to the above-mentioned mechanisms in that the cost of use is very low when copying several files. The results from the user study suggest that PDD is a superior alternative as most users would prefer using it to existing alternatives (stated by 25 out of 32 participants). Perhaps this is just because of the novelty of the interaction technique, however, we may also consider other factors such as the pervasiveness of cell phones, which makes them a more natural way to transfer files than flash drives or other mechanisms. Three of the seven participants who preferred other mechanisms are also three (out of seven) users who do not own a smartphone.

Post-experiment analysis of the recorded videos and received feedback allowed us to understand why some users were unsuccessful at transferring all files, which gives us room to improve aspects of the PDD interface. The main reasons users had trouble with PDD were four: frustration, comfort, focusing and distance, and others, which are explained below.

- **Frustration:** The mobile user is a demanding and fast user. PDD takes between 5-10 seconds to capture the image of a file to be transferred. This seems like a lot of time for mobile phone users and may cause some frustration.

- **Comfort:** The application requires the user to hold the phone with one hand and transfer files with the other. Since the experiment used an iPhone4, it appeared to be heavy for some users' hands, which made capturing the picture difficult because their hands shook and they lost focus. This can be solved in two ways: first, we may use newer versions of iPhone (an iPhone 5 is 18% lighter than iPhone 4), and we may incorporate algorithms to minimize shakiness in PDD.

- **Focusing and distance:** The optimal distance between smartphone and screen for capturing the file is approximately 10-15 cms. However, this was not immediately apparent to some users before transferring the file, so they had more trouble finding the ideal distance and capturing the file (some tried to do it e.g. by filling the entire iPhone screen with the file icon). This item affects directly the effectiveness, usability and user perceptions about PDD.

- **Others:** Other issues that were not foreseen by us and present interesting opportunities for improvement are the following: one user thought his fingers were too big to use PDD because it was hard to transfer files to the sidebar, awareness about transferred files could be improved, and multi-finger selection of several files could also be incorporated. Also, file and background detection need to be improved, to neutralize the effects of shakiness and focusing distance.

Another area for improvement is the fact that the file must be open in the desktop computer for it to be copied to the smartphone device. This is naturally costly, since the user must deal with the mouse to find and open the correct folder, and then begin the process with the smartphone. Though we note this problem is similar in other transfer techniques (e.g. flash drives), it may be improved by incorporating navigation through folders in PDD.

Finally, it is important to comment that the group of participants who evaluated PDD was somewhat homogeneous in age, gender and occupation (all were 18 to 35 years old, 81% were male, all were undergraduate or graduate university students). This may influence acceptance rates, as young people with some knowledge in technology are probably more positive towards new interaction techniques. Therefore, it is important to continue evaluating this application in more heterogeneous groups to examine how to improve it to make it intuitive for other types of users.

7 Conclusions

This work proposed an augmented-reality-based interaction technique for file transfer in multiple computer environments. Specifically, we provide a new interaction technique for multiple file transfer in non physically co-located displays. This technique may be easier to use and more natural than traditional techniques, due to the pervasiveness of cellphones and due to the fact that the proposed technique uses several well-known interaction techniques (drag and drop, tapping/clicking on files) and smartphone capabilities (applications, phone camera).

The results from a user evaluation study are encouraging. 78% of participants stated they preferred PDD to their usual alternatives, and we also received encouraging comments during the evaluation (e.g. asking when the application will be commercially available in the Apple App Store). However, during the second stage of the experiment, we only had users compare PDD to using a pendrive, so the opinion of the participants about other techniques is from their own personal experience, as the experiment did not allow them to test all of the existing alternatives. It is also important to note that the testing group consisted of young engineering students, who may be more open to this kind of application, and further testing is required by other user groups. However, we believe the positive results are not only because of these aspects, but also because the application is fun and novel to users, and truly allows efficient file transfer from computer to phone, from phone to computer and between computers, for several files and over any distance. Naturally, this requires involved computers to have pre-installed PDD server capabilities, but the server is very lightweight and needs only to be installed once at setup.

As discussed earlier, the PDD application is a prototype used to test the proposed interaction technique; it needs further work to become an effective, completely usable application in real-world settings. One challenge in this area is improving the wallpaper detection: currently, the application detects from a small number of plain background colors to ease the detection process and focus on testing the interaction technique. Another challenge is detecting not only the wallpaper, but also if a screensaver is on, to deactivate it and then begin with the file transfer process (currently, the user must deactivate the screensaver manually before using PDD). It is also important to consider testing the application outside of a laboratory, in settings in which lighting is not controlled.

An interesting area of future work is to study more complex scenarios, in which several users exchange files, similar to the scenario of interaction with large displays proposed in [Boring et al., 2011]. In this case, there is a multiple computing device scenario but the devices belong to more than one user. For example, a file sharing scenario for this case are meetings in which several attendees want to present information on one projector. Although the current implementation supports this scenario, it does not provide security or authentication so any user could copy and transfer files to any other user. Another dimension of future work is using AR not only for copying files, but for providing an additional information layer about the files, e.g. size, owner, or type of file.

Acknowledgements

The authors would like to thank the volunteers who participated in the experiment, as well as acknowledge the help of Ignacio Solís, Katherine Paez and Carlos Meneses in the organization of the experiments and providing the figures for the present article. We would also like to thank the reviewers for their insightful comments.

This project was partially supported by Fondecyt project No. 11110056, and the Conicyt PhD scholarship.

References

[Azuma, 1997] Azuma, R. (1997). A Survey of Augmented Reality. Presence *6*, 355–385.

[Boring et al., 2010] Boring, S., Baur, D., Butz, A., Gustafson, S. and Baudisch, P. (2010). Touch projector: mobile interaction through video. In CHI pp. 2287–2296,.

[Boring et al., 2011] Boring, S., Gehring, S., Wiethoff, A., Blöckner, A. M., Schöning, J. and Butz, A. (2011). Multi-user interaction on media facades through live video on mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '11 pp. 2721–2724, ACM, New York, NY, USA.

[Chang and Li, 2011] Chang, T.-H. and Li, Y. (2011). Deep Shot: A Framework for Migrating Tasks across devices using mobile phone cameras. In Proceedings of CHI 2011.

[Guttman, 2001] Guttman, E. (2001). Autoconfiguration for ip networking: Enabling local communication. Internet Computing, IEEE *5*, 81–86.

[Herbert et al., 2011] Herbert, L., Pears, N., Jackson, D. and Olivier, P. (2011). Mobile Device and Intelligent Display Interaction via Scale-invariant Image Feature Matching. In PECCS pp. 207–214,.

[Mistry et al., 2011] Mistry, P., Nanayakkara, S. and Maes, P. (2011). SPARSH: touch the cloud. In Proceedings of the ACM 2011 conference on Computer supported cooperative work CSCW '11 pp. 585–586, ACM, New York, NY, USA.

[Nacenta et al., 2005] Nacenta, M. A., Aliakseyeu, D., Subramanian, S. and Gutwin, C. (2005). A comparison of techniques for multi-display reaching. In Proceedings of the SIGCHI conference on Human factors in computing systems CHI '05 pp. 371–380, ACM, New York, NY, USA.

[Rekimoto, 1997] Rekimoto, J. (1997). Pick-and-drop: a direct manipulation technique for multiple computer environments. In Proceedings of the 10th annual ACM symposium on User interface software and technology UIST '97 pp. 31–39, ACM, New York, NY, USA.

[Rekimoto, 2000] Rekimoto, J. (2000). Multiple-computer user interfaces: "beyond the desktop" direct manipulation environments. In CHI '00 extended abstracts on Human factors in computing systems CHI EA '00 pp. 6–7, ACM, New York, NY, USA.

[Schmidt et al., 2012] Schmidt, D., Seifert, J., Rukzio, E. and Gellersen, H. (2012). A cross-device interaction style for mobiles and surfaces. In Proceedings of the Designing Interactive Systems Conference DIS '12 pp. 318–327, ACM, New York, NY, USA.

[Seewoonauth et al., 2009] Seewoonauth, K., Rukzio, E., Hardy, R. and Holleis, P. (2009). Touch &; connect and touch & select: interacting with a computer by touching it with a mobile phone. In Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services MobileHCI '09 pp. 36:1–36:9, ACM, New York, NY, USA.

[Srivastava, 2005] Srivastava, L. (2005). Mobile phones and the evolution of social behaviour. Behaviour & IT *24*, 111–129.

[The Nielsen Company, 2011] The Nielsen Company (2011). Factsheet: The US Media Universe. `http://blog.nielsen.com/nielsenwire/online_mobile/factsheet-the-u-s-media-universe/`, accessed September 2011.

[van Krevelen and Poelman, 2010] van Krevelen, D. W. F. and Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. The International Journal of Virtual Reality *9*, 1–20.