What is that? Gesturing to determine device identity

Colin Swindells
School of Computing Science
Simon Fraser University
Burnaby, BC V5A 1S6 Canada
+1 604 291 4371
swindell@sfu.ca

Kori M. Inkpen
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5 Canada
+1 902 494 2093
inkpen@cs.dal.ca

ABSTRACT
Computing devices can seamlessly recognize one another as they join and leave a wireless network, but users often experience difficulty identifying a desired device from a continuously changing list of devices surrounding them. This paper describes our custom implementation of a stylus and tags that enable users to rapidly identify devices in dynamic environments. Our system utilizes a natural pointing gesture to identify a device, and subsequently transfer data over a wireless network.

Keywords
Pointing, gesturing, face-to-face, collaboration, handheld computer, PDA, infrared (IR) tag

INTRODUCTION
As short-range wireless technologies such as 802.11 and Bluetooth become widespread, our environments are becoming littered with computing devices that can automatically identify each other. However, as the number of devices grows, it becomes increasing difficult for *users* within these environments to quickly identify a particular device. Rapid identification of computing devices, such as handhelds, laptops, or phones, is an essential requirement for people who use these devices to seamlessly exchange ideas and information. Thus, while new advances in wireless technologies help to create seamless computer–computer identification, human–computer identification is becoming increasingly difficult.

Often when we interact in short-range wireless networks, we can physically see the device with which we want to interact (e.g. “that one there”). Searching through a hierarchy of device names seems counter-intuitive when the device is sitting right in front of us! As suggested by Bolt [1], gesturing in conjunction with a graphical interface is more natural than typing symbols. We describe an implementation of a line-of-sight, tag-based identification system. By allowing users to identify devices by simply pointing at them, this system facilitates data transfer in a wireless network.

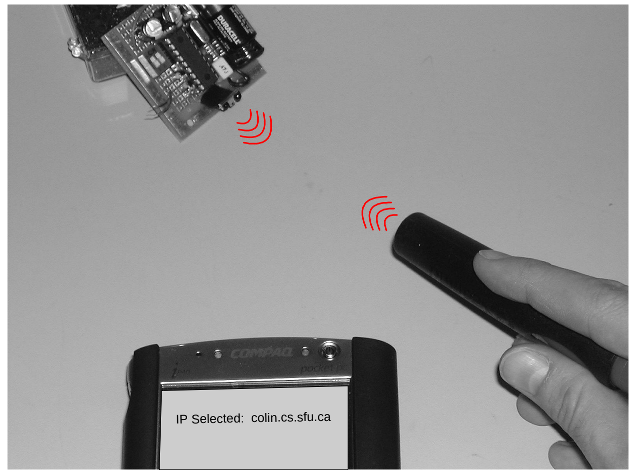

Figure 1: Line-of-sight identification with tag and stylus

Example scenario: sharing a digital photo album
A friend wants to give you a copy of a digital photo album that is stored on their handheld computer. Your friend must first identify your computer and then transfer the photo album. To perform these steps, your friend could select the photo album on their handheld, then point to your laptop’s infrared port, and ‘beam’ the data (i.e. send using line-of-sight IR communications). ‘Beaming’ is advantageous because the target device is implicitly identified by pointing towards it. However, beaming is problematic because both devices must remain stationary with no disruption to the line-of-sight infrared connection during the complete data transfer. In practice, this motionlessness is quite awkward, particularly for large files. In addition, data can only be transferred between two devices at any given point in time.

Alternatively, your friend could transfer the album by identifying your laptop (e.g. its IP address), and copying it over a wireless network. By separating identification from data transfer, we resolve the two major disadvantages of ‘beaming’ mentioned above. Unfortunately, with current systems, the user must typically select the device from a user interface. As a result, identifying the target device is not as direct as pointing towards it.

Our solution: identification via pointing
Pointing towards a target computing device can be easier and more natural than selection from a list-box widget of devices or entering the device’s IP address into a text box. In particular, selecting a target device from a list of

networked devices is a daunting task when the environment contains dozens or even hundreds of devices – many with non-descriptive names. Once a device has been identified through a pointing gesture, any necessary data transfer can then be carried out over the wireless network.

This solution combines the benefits of infrared and wireless communication. Using a line-of-sight technology such as infrared for the pointing, we can uniquely identify a device through a gesture. Then, once the device has been identified, users are free to move about and work on other tasks while data is transferred over the wireless network.

The remainder of this paper summarizes related literature, our implementation of a custom tag-based system for rapidly identifying objects, and our on-going work.

RELATED WORK

Similar technologies include Ulmer and Ishii's mediaBlocks [3], and HP Lab's E-Squirt technology [2]. mediaBlocks embed digital ID tags in physical blocks to function as containers for digital media. These 'blocks' can capture, transport, and retrieve online digital media. E-squirt technology uses small infrared transmitters to "squirt" URLs into devices. The device then displays the appropriate multimedia content retrieved from the Internet.

IMPLEMENTATION

To identify a target with a pointing gesture, we developed a custom stylus that can be used with an IPAQ™ handheld computer. We added an IrDA (InfraRed Data Association) compliant infrared transceiver to the stylus, and then prototyped tags that can be fixed to other devices, walls, etc. and communicate with the stylus. The tags are stand-alone devices based on tags developed by Poor [1], and the stylus needs only one serial communications link between itself and the handheld. This solution maximizes device, platform, and application independence. Furthermore, tags can be placed on inanimate objects that are logically related to a computing device but not physically connected to the device (e.g. a projected wall display). The infrared transceiver technology we used supports link distances of at least 1.5 m, whereas most current transceivers in handhelds and laptops fail beyond 0.3 m.

Figure 2 illustrates the communication flow from the stylus to a tag. The steps required to identify a device are:

1. The user points the stylus towards a tag, and presses a button on the stylus. A microcontroller within the stylus detects the button event and pings the tag using its infrared transceiver.
2. The tag's microcontroller receives the ping message, and then sends its identity information (e.g. IP address of the device) back to the stylus.
3. The stylus receives the identity information and checks the message validity with a cyclic redundancy check.
4. If the identity information from the tag is correct, it is sent via a standard serial communications link to the target device (e.g. handheld).

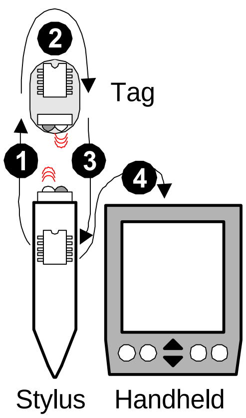

Figure 2: Communications Flow

CONCLUSION AND FUTURE WORK

As people join and leave the range of a wireless network, their devices become visible and invisible, respectively, to others in the network. Consequently, any list of devices within the network will be continually changing. Our pointing method will help ensure that these changes are transparent to the user.

We are currently conducting a formal user study to compare the identification of computing devices using our pointing method to the use of traditional user interface widgets such as a list. Preliminary results from our pilot tests suggest that identifying a device from a list is straightforward if the device is physically labeled and the list is short (< 5 items). With lists of 10+ items, scrolling and selection takes longer and requires more concentration – especially on handheld displays. If the target device is not labeled, the user must spend additional time finding the device's ID (e.g. by asking someone or searching the target device's network settings).

Our future research will also address overall usability of the system, such as transmission collisions, in environments with varied densities of computing devices. Additionally, varying the infrared link distance and beam angle will also impact the usability of our pointing method.

ACKNOWLEDGMENTS

We thank Melanie Tory and the members of the EDGE Lab for their helpful suggestions. We are also grateful for the financial support from Colligo Networks, Inc., the BC Advanced Systems Institute, and Simon Fraser University.

REFERENCES

1. Bolt, R. (1980) Put-That-There: Voice and Gesture at the Graphics Interface. Proc. of SIGGRAPH '80, pp. 262-270.
2. Bucholtz, C. (2000) Software Behind HP Labs' CoolTown. Available at: http://www.interex.org/hpworldnews/hpw005/03news.html
3. Ullmer, B., and Ishii, H. (1999) mediaBlocks: Tangible interfaces for online media. Proc. of CHI '99, pp. 31-32.
4. Poor, R.D. (1999) The iRX 2.1. Available at: http://www.media.mit.edu/~r/projects/picsem/irx2/