Multi-State Device Tracking for Tangible Tabletops

Ali Alavi[1], Brice Clocher[1], Allen Smith[1], Andreas Kunz[2], and Morten Fjeld[1]

[1]t2i Lab, Chalmers TH, SE-41296 Gothenburg, Sweden
[2]ICVR, ETH Zurich, CH-8092 Zurich, Switzerland

Abstract

On tangible tabletops, Tangible User Interfaces (TUIs) can signalize their identity, position, orientation, and state by active infrared light. This provides rich interaction capabilities in complex, dynamic scenarios. If TUIs have to transfer additional high-resolution information, many bits are required for each update. This has a negative impact on the overall update rate of the system. In the first part of this paper, we present an in-house map application where interaction with time-dependent contour lines may benefit from high-resolution TUI states. Prototypical TUI concepts such as slider, ruler, and dials further motivate the benefit of high-resolution tracking. In the second part of the paper, we depart from a device tracking overview and then show how tangible devices for tabletops typically use infrared (IR) emitters and a camera to send information about their position, orientation, and state. Since transferring many additional information bits via a normal camera-based tabletop system is not feasible anymore, we introduce next a new system setup that still offers a sufficiently high update rate for a smooth interaction. The new method can be realized as a tabletop system using a low-cost camera detecting position, combined with a low-cost infrared receiver detecting the state of each device. Since both kinds of sensors are used simultaneously we call the method "dual mode." This method combines a camera-based tracking with the possibility to transfer an almost unlimited amount of states for each device.

Categories and Subject Descriptors (according to ACM CCS):
General terms: Algorithms, Performance
Keywords: Active tangible devices, tabletop, dual mode, IR tracking, multiple TUI

1. Introduction

The advent of widely available interactive tabletops has created high expectations among users for such systems. Tangible tabletops where active devices (TUIs) are tracked to inform about their identity, position, orientation, and state can provide rich interaction within complex, dynamic scenarios. InfrActables [GSK06] delivers such an experience by providing users with active TUIs. In that system, TUIs have form factors such as pen, handle, ruler, and color tool, which may enable a more natural interaction style. Each of these TUIs has a few state-triggering widgets, such as buttons on top of the bricks or a micro switch under the pen's tip. Thereby, users can control states while positioning the TUI on the surface. Informing about TUI states over a large range at a high-resolution requires sending many bits for each update and comes at the cost of system update rate. However, tracking of TUIs has so far been limited to the spatial and temporal resolution of the camera CCD chip. Future scenarios where TUIs control dynamic high-resolution parameters of dynamic map-based scenarios, economic simulations, or science education are promising [KF10]. In such application, TUIs with dynamic high-resolution input streams will enable system designers to use richer forms of interaction. For instance, a slider, ruler, or dial equipped with a potentiometer can send its input value (sampled though an A/D converter) offering up to 2048 adjustment levels. To implement such functions for systems like InfrActables, a TUI needs to employ a much higher number of state bits than today. We could not utilize this number of states on the initial system because the update rate would then drop dramatically. In this paper, we address this problem by introducing a new additional state detection method for use in active tangible tabletops. This method allows us to build tangible tabletops with a high number of states using low-cost components.

The first part of this paper presents an existing map-based application where interaction with time-dependent contour

Figure 1: *A tangible tabletop map-based application for TUI and pen-based interaction with Google Maps.*

lines may benefit from high-resolution TUI states. Previously researched TUIs such as slider, ruler, and dials can be expected to work with our system. Besides being a research contribution in and of itself, our application serves also as motivation to research improved multi-state device tracking. In the second part of this paper, we first give an overview of alternative forms of tabletop tracking before we focus on the tracking of actively emitting devices. The second contribution of this paper, an improved multi-state device tracking method, called "dual mode", is then presented. The paper summarizes the current status and indicates potential future work.

2. Tangible Tabletop Map-based Application

Users involved in time-critical planning with interactive maps sometimes use physical tools like ruler, dials, and pens in order to share knowledge with each other and collaborate in creating a common operational picture. Previous research has demonstrated that tangible tabletops can help in these tasks [PHF09]. For the purpose of crisis resource management, we have built an interactive table with tangible devices and an information visualization framework prototype. The framework allows creating crisis management scenarios using Google Maps-based Flash applications. To interact with this application, we propose actively emitting TUIs such as slider, ruler, and dials. While the design and use of these devices has been proposed in related projects [GSK06] [SJG*06] [WWJ*06], we consider the tracking method presented next as critical in making the use of such devices successful.

One of the most important features of a tangible crisis management application is believed to be time-dependent shape, or contour line, editing. Indeed, crisis management experts often draw shapes on paper maps, for instance, in order to represent the spreading of a fire. Those shapes are

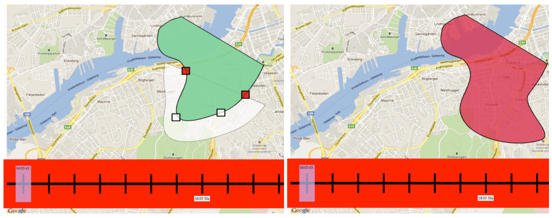

Figure 2: *Map interaction screenshot: Shape control is done by moving the nodes of a parametric curve; time control employs time-line at the bottom.*

associated with thematic, spatial, and temporal content. In the case of a fire, a shape may represent an area that is burning at a specific time. In one related example, Igarashi et al. [IMH05] presented algorithms and applications where users can move and deform a two-dimensional shape without manually establishing freeform deformation (FFD) domain beforehand. Inspired by this work, one of our current implementations employs a so-called parametric curve description. Using Flash parametric curve libraries, we have been able to implement from a rather high level of abstraction (Figure 2). When expert users receive data about the development of a phenomenon, they often model this development using time-dependent shapes. A first problem encountered when drawing shapes on a map is how to associate temporal and thematic content with a shape. In our system this raises two issues: firstly, how tangible interaction can make shape and time control precise but still easy and intuitive, and, secondly, how a user can modify shapes while still being able to keep track of their associated temporal, spatial, and thematic contents. As for the first issue, we conjecture that TUIs such as slider, ruler, and dials may benefit shape and time control. Figure 3 shows some prototypical uses of multi-state devices such as dials and a frame. Tracking TUI states with values assuming values over a large range at a high-resolution requires sending many bits for each update without compromising the system update rate. Achieving this without making compromising on system update rate is the focus for the remainder of this paper.

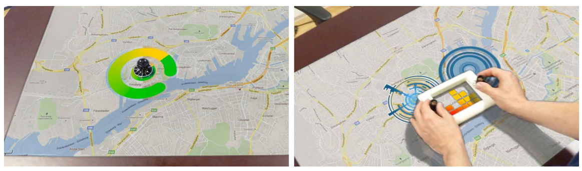

Figure 3: *Multi-state device on existing tabletop system showing tangible continuous parameter control for values such as time, radiation level, or population (left); selection frame with two-handed continuous control of radar visualizations or mode selectors (right). (Simulated images)*

2.1. Device tracking

Since a tabletop system is both a display and a multitouch input device at the same time, the complementary use of physical devices providing input may benefit certain types of applications [SR09]. Such devices may improve the fluidity and reduce the cognitive load of the user-system interaction [FB09]. Thus, the tabletop system must be able to distinguish between intended input from devices, and unintended input from other objects on the table [KF10]. Furthermore, tabletop systems must be able to detect multiple devices simultaneously when one or multiple users are interacting with the system.

2.2. Tracking in Tabletop Systems

In order to enable intuitive interaction with the content visible on the tabletop, devices other than the mouse and keyboard must be used. There is a class of devices that are easily identifiable by their inherent function known as 'physical icons' or phicons [IU97]. In this case, each device usually has a static association so that the tabletop system is able to detect its identifier (ID) in addition to its position. Once the device's ID is known to the system, the underlying functionality is also defined since the association cannot be changed. However, other tabletop devices might have a dynamic association that allows for simpler detection algorithms. In the latter case, the devices have a more general character, and so the intuitiveness is only guaranteed by the displayed content, i.e., the graphical user interface (GUI). The dynamic association is user-triggered and follows predefined steps. These steps may require some learning on the part of the user.

Tabletop systems must be able to detect the position of an interaction device and, in the case of phicons or other specialized input devices, their ID. While it is important in a global context for the position of the device to be displayed on the tabletop's surface, the ID is relevant for integrating a device's specialized functionality into a specific application. More degrees of freedom (DOF) than given by planar interaction become relevant. For instance, the z-coordinate may be used to distinguish between writing and pointing in pen-based interaction. Additionally, the tracking and detection system's latency should be below the user's perceptual threshold, otherwise user irritation may occur. During normal operation on a tabletop system, various objects may be placed on the surface which are not meant for interaction, but which could interfere with the system, e.g. by shadowing effects. Unlike a mouse, which is a relative pointing device that detects the travelling distance and orientation, all tracking systems for tabletop systems allow absolute pointing: the object is detected at precisely the place where the user puts the device.

2.3. Tracking Active Devices

Our work has been inspired by two contributions in the area of tangible tabletops using active TUIs: SmartFiducial [HK11] and MightyTrace [HKK08] [Hof11]. In SmartFiducial, active tangibles communicate with a host computer using wireless radio frequency (RF) transmission. Object positions are detected using a visual tracking system, thus adding a further level of complexity to the TUI design, as well as the host computer. Moreover, there is always a risk of unintended interference with RF communication. Hence, we deemed this approach to be unsuitable for our application. Inspired by the fast IR-sensors in MightyTrace, we decided to apply them as additional sensors to a camera-based system. In MightyTrace, a matrix of IR sensors is used to detect the position and state of TUIs. Each device is assigned a specific time frame during which its LED is turned on. Thus, each device can be detected unambiguously. For sending states, there are even more time slots for each device. For example, if a device has eight different states it then needs four time slots, one for detecting its position and three for sending its state bits (thus eight different states). However, the main restriction with such a time-multiplexed approach is that the system update rate is limited not only by the number of devices on the surface, but also by the number of states each device may have.

2.4. Dual Mode Tracking Method

Here, we introduce a new approach to improve multi-state IR tracking of tangibles on tabletops. Mainly, we aim to design low-latency active devices, that is, devices with more states without reducing the high refresh rate of the system. Moreover, we are interested in a cost-effective solution. We suggest a distinct way to combine a low-cost camera for position detection and a low-cost IR receiver for state detection of each device, in what we call a dual mode approach.

Considering the physical size of a table and the states required for tools such as sliders, rulers, and dials, we may simultaneously have five devices on the surface, each with 2048 possible states or adjustment levels. To meet this requirement, we use two different receivers: a camera capable of detecting positions, and an IR receiver capable of detecting the states of the devices. Concerning latency requirements, two terms are frequently employed: "update rate," which is the number of positions and states being updated per second, and "lag," which is the "response time" of the system to user input.

Much like in QualiTrack [HNK09], the devices and the camera are synchronized. Moreover, each device is assigned to a specific time slot. All devices emit IR light on each frame, except on their assigned time slot (Figure 4). Since the camera sees all the devices in every frame except one, the average update rate of the system equals f (M-1)/M, where f is camera frame rate (Hz) and M is the maximum number of devices we want to have on the table (which is preconfigured and constant in the system). Camera frames are indexed in cycles of M frames. Since newly placed devices wait for up to one cycle to start IR transmission, the maximum setup delay equals M/f. Thus, using the dual mode method, the number of devices does not reduce the system's update rate, nor does it increase its lag. Only the setup delay will be negatively affected. We can now unambiguously identify each device and its position. Device tracking uses a blob-tracking algorithm [SHB*10]. While we assume that a device has one LED source only, instrumenting a device with two or more sources and combining their positions can give device orientation [Hof11].

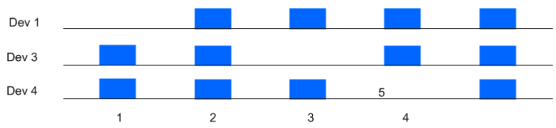

Figure 4: *Position information transfer example with M set to 5 and with devices 1, 3, and 4 (rows) present. A cycle of 5 frames (columns) is shown. Blue cells are IR flashes sent by the devices to be detected by the camera.*

Each device transmits its state information using its IR LED between two synchronization signals, i.e., the speed of transmitting the state information is significantly higher than the speed of the camera (Figure 5). This is feasible since the state information is read by a simple IR receiver and not by the camera. The interval between two consecutive camera frames is further divided into M sub-frames. Within each sub-frame, only the corresponding device sends its state information. The bit rate, R, of the sensor we employ can be up to 22 kbps. Hence, with the camera exposure time, e, and f and M as defined above, the maximum number of state bits per device equals: R ((1/f - e)/M). For example, with M set to 5, f at 60 Hz, and e at 10 ms, each device can transmit up to 29 bits of state information, allowing more than half a billion states per device.

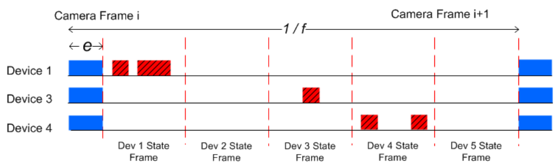

Figure 5: *State information transfer example with M set to 5 and with devices 1, 3, and 4 (rows) present. One interval between camera frames i and i+1 is shown. Blue cells are IR flashes to be detected by the camera; red cells are IR flashes to be detected by the IR receiver.*

2.5. Current status and future works

At an application level, we plan to further examine tasks requiring the use of multi-state tangible devices on tabletops. To this end, we plan to realize the slider, ruler, and dials accompanying the pen for use with our map-based tabletop application. Drawing on design principles and solutions from previous work [HKK08] [HNK09] [GSK06], this will call for engineering new TUIs tailored to this use. In a later phase, we foresee designing, running, and analyzing user studies to validate the usability and acceptance of the solutions.

At a tabletop device tracking level, we have evaluated the feasibility of the proposed method by implementing its essential subsystems. In particular, we implemented the IR receiver and changed the QualiTrack TUIs to send state information using our dual mode method. We also investigated whether the battery operated TUIs allow us to send signals powerful enough to be detected by the IR receiver, considering the distance between them. Our findings show that it is feasible to implement a tabletop using our new method. A next step in this project will be to implement a complete tangible tabletop using the method with the suggested slider, ruler, and dials.

2.6. Acknowledgements

We greatly thank Erik Tobin for improving the readability of this text.

References

[FB09] Fjeld M., Barendregt W.: Epistemic action: A measure for cognitive support in tangible user interfaces? *Behavior Research Methods 41*, 3 (2009), 876–881.

[GSK06] Ganser C., Steinemann A., Kunz A.: Infractables: Multi-user tracking system for interactive surfaces. In *Proc. of the IEEE conference on Virtual Reality (VR '06)* (2006), pp. 253–256.

[HK11] Hochenbaum J., Kapur A.: Adding z-depth and pressure expressivity to tangible tabletop surfaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (Oslo, Norway, 2011), Jensenius A. R., Tveit A., Godøy R. I., Overholt D., (Eds.), pp. 240–243.

[HKK08] Hofer R., Kaplan P., Kunz A.: Mighty trace: Multiuser technology on lcds. In *Proc. CHI '08* (2008), pp. 215–218.

[HNK09] Hofer R., Nescher T., Kunz A.: Qualitrack: Highspeed tui tracking for tabletop applications. In *Human-Computer Interaction, INTERACT 2009*, Gross T., Gulliksen J., Kotz P., Oestreicher L., Palanque P., Prates R., Winckler M., (Eds.), vol. 5727 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, 2009, pp. 332–335.

[Hof11] Hofer R.: *Tracking technologies for interactive tabletop surfaces*. PHD Thesis, ETH Zurich, 2011.

[IMH05] Igarashi T., Moscovich T., Hughes J.: As-rigid-as-possible shape manipulation. *ACM Trans. Graph. 24*, 3 (July 2005), 1134–1141.

[IU97] Ishii H., Ullmer B.: Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proc. CHI '97* (1997), pp. 234–241.

[KF10] Kunz A., Fjeld M.: From table-system to tabletop: Integrating technology into interactive surfaces. Human-Computer Interaction Series. 2010, pp. 53–72.

[PHF09] Piazza T., Heller H., Fjeld M.: Cermit: Co-located and remote collaborative system for emergency response management. In *Proc. SIGRAD 2009* (2009), pp. 12–20.

[SHB*10] Schoening J., Hook J., Bartindale T., Schmidt D., Oliver P., Echtler F., Motamedi N., Brandl P., Zadow U.: *Building Interactive Multi-touch Surfaces, Tabletops - Horizontal Interactive Displays*. Springer, 2010.

[SJG*06] Shahrokni A., Jenaro J., Gustafsson T., Vinnberg A., Sandsjo A., Fjeld M.: One-dimensional force feedback slider: going from an analogue to a digital platform. In *Proc. NordiCHI '06* (2006), pp. 435–456.

[SR09] Shen C., Ryall K. e. a.: Collaborative tabletop research and evaluation: Interfaces and interactions for direct-touch horizontal surfaces. In *Interactive artifacts and furniture supporting collaborative work and learning*, Dillenbourg P., Huang J., Cherubini M., (Eds.). Springer Science and Business Media, New York, Oslo, Norway, 2009, pp. 111–128.

[WWJ*06] Weiss M., Wagner J., Jansen Y., Jennings R., Khoshabeh R., Hollan J., Borchers J.: Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *Proc. CHI '09* (2006), pp. 481–490.