TViews: an extensible architecture for developing multi-user digital media tables

Ali Mazalek
Assistant Professor
Synaesthetic Media Lab, Georgia Tech, Atlanta GA
mazalek@gatech.edu

Matthew Reynolds
Co-Founder
ThingMagic, Cambridge MA
matt@thingmagic.com

Glorianna Davenport
Principal Research Associate
Media Fabrics Group, MIT Media Lab, Cambridge MA
gid@media.mit.edu

TViews: an extensible architecture for developing multi-user digital media tables

Abstract

As digital entertainment applications evolve, there is a need for new kinds of platforms that can support shared media interactions for everyday consumers. In recent years this has resulted in the development of several technical approaches to digital media tables, but none of these early attempts point to a general purpose, economically viable media table system.

In this paper we present TViews, an extensible method and acoustic-based sensing framework for creating digital media tables that are designed to overcome the limitations of single purpose systems. The system provides a means for real-time multi-object tracking on the tabletop and for the management of large numbers of objects and applications across multiple platform instances. The objects can be physically customized to suit particular applications, and the table provides output via a coincident embedded display. By providing a general purpose solution for media table development, we give application designers the freedom to invent a broad range of media interactions and applications for everyday social environments, such as homes, classrooms and public spaces. We discuss our prototype applications for digital media browsing and game play, examining different approaches to mapping physical interaction objects to the media space as either generic controls or fixed function devices.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces---*input devices and strategies, interaction styles*; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces---*collaborative computing, synchronous interaction*; H.2.8 [Database Management]: Database Applications---*image databases*

Keywords

Media table, object sensing, extensible architecture, tangible interaction, multi-user platform, collaborative, media applications.

The widespread appropriation and use of tables across different cultures and throughout history suggests that tables provide an ideal space for people to engage in shared and sociable interactions during many different kinds of activities. In our increasingly digital age, the development of media table platforms has the potential to extend these tabletop activities into the digital realm, shifting human-computer interactions away from today's dominant desktop-based paradigm to a broader one of shared physical contexts in everyday spaces such as home living rooms, school classrooms, and workplace conference rooms, as well as public spaces such as cafes, shops or museums. To accommodate these different physical spaces and the correspondingly different activities that take place within them, there is an increasing need to develop a general purpose interactive tabletop display surface that can support a broad range of media applications, such as media asset management, story construction, multimedia learning, and digital game play, as well as many other applications that will be found by the application developer community that we hope will arise to support media table development.

At its most basic level, a media table needs to provide a horizontal interaction surface upon which the spatial configuration of tagged objects can be computationally interpreted and then subsequently augmented with coincident visual output. In existing prototypes, the visual output is usually provided by rear or front projection, but the rapid development of large plasma and LCD displays at consumer price levels makes these display technologies increasingly attractive options for media tables. While some interactive table displays have made use of touch as a means of input rather than object tracking, the additional benefit provided by object tracking is the ability to identify and associate unique physical artifacts with different elements or functions within the application space. These physical artifacts can take on a form that is representative of their digital functionality, thus allowing some of the information within the interactive environment to be off-loaded from a purely graphical form to a hybrid graphical and physical-world representation.

Early work on interactive tables began with research prototypes such as the Digital Desk, built at Xerox EuroPARC, and the ActiveDesk work at the University of Toronto. The Digital Desk made use of an ordinary physical desk with a video camera mounted above that could track the movements of an LED tipped pen to determine where the user was pointing [Wellner 1993]. Later versions included overhead digital projection of electronic objects onto regular paper documents. The ActiveDesk allowed two-handed input through physical artifacts called "bricks" and was used with a drawing application called GraspDraw [Fitzmaurice 1995].

The next major steps in interactive tabletops took place at the MIT Media Lab, with the metaDesk and I/O Bulb projects [Ullmer 1997, Underkoffler 1999]. These systems differed from the Digital Desk and ActiveDesk systems in that they were designed for networked collaborative use. Applications included geographical visualization, simulation of holographic setups, and urban planning. In both systems, the tracking of multiple objects was achieved through computer vision, and the display was accomplished with projected graphics from the rear (metaDesk) or the front (I/O Bulb). Today, interactive tables are being explored for different applications in a variety of physical contexts. Notable examples include the DiamondTouch table from MERL which allows multiple users to interact simultaneously via touch [Dietz 2001] and the Sensetable project from the MIT Media Lab which is based on Wacom's tablet and pen technology [Patten 2001]. Wacom tablets use an antenna grid within the sensor board to track pens containing coil-and-capacitor resonant circuits. While a typical Wacom tablet can track only two pens at a time, the Sensetable developers modified the system using a duty cycling approach in order to allow a greater number of objects to be tracked at once.

While the past decade has yielded an increasing number of research efforts to develop media tables, it is important to note that the technical approaches employed have not yet pointed to a general purpose, economically viable tabletop display and interaction platform. In particular, existing approaches have typically been designed with a single application area in mind and have not succeeded in providing a scalable architecture that can support a diverse range of media applications that would be required for everyday use by many people across many different contexts. In this paper we introduce TViews, an extensible method and sensing framework for developing multi-object tracking media tables that can manage large numbers of tracked objects while running many different media applications at once. The difference in approach is illustrated in Figure 1. In the following sections, we describe the TViews media table architecture and our initial application development in the areas of media content organization, geographic media browsing, digital game play, and media improvisation.

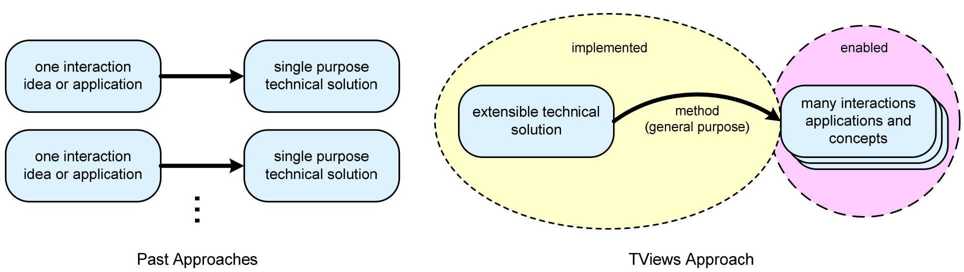

Figure 1:

TViews approach contrasted with past approaches to media table technology design.

TViews System Design Goals

The TViews table is a media interaction platform designed for shared living spaces within the home environment. This platform consists of three parts: a dynamic object tracking and identification system that works with an extensible set of tagged objects on the table's surface, a planar display that is integrated into the table's surface, and a set of application programming interfaces (APIs) that tie these elements together. TViews applications can therefore provide visual output that coincides with the movements of the tagged objects as well as the actions signaled by the human manipulation of those objects. Figure 2 illustrates the physical configuration of the TViews table.

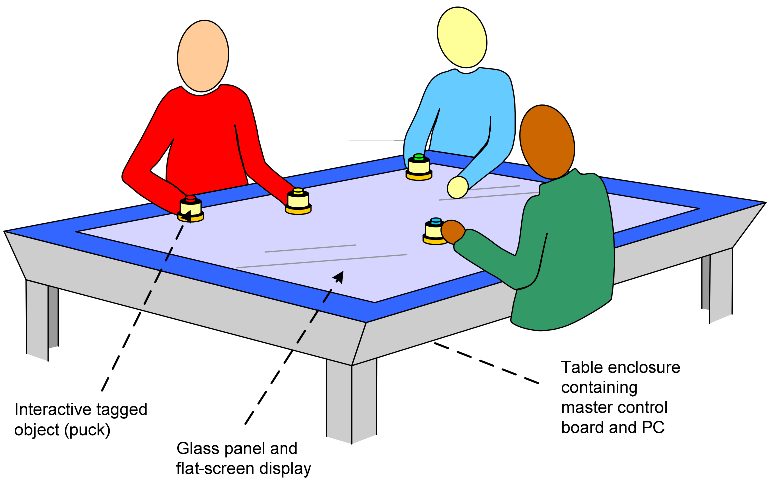

Figure 2:

Physical configuration and setup of the TViews media table.

In our exploration of the use of interactive objects on a tabletop display as a platform for shared media and story interactions [Mazalek 2003], we identified a number of requirements necessary for the development of an extensible architecture for media tables:

(a) Scalability of the display and interaction space in size.

(b) Extensibility of the object namespace to accommodate a virtually unlimited number of unique objects.

(c) Portability of interactive objects between platforms.

(d) Extensibility of the application space and management of objects and applications.

TViews utilizes a novel combination of acoustic sensing and infrared communication technologies coupled with distributed processing both in the interaction objects themselves as well as in the table's central processor to achieve these objectives. We will first provide a quick summary of the combination of technical features that have been used to fulfill these requirements in the TViews system.

In order to allow **scalability** of the interaction surface in size, it is important that the TViews object positioning technology function in a manner independent of the size and scale of the interactive surface. Thus a TViews surface may be constructed in smaller or larger versions and at different scales and aspect ratios to accommodate different kinds of displays. One way in which this can be accomplished is through a time-of-flight approach, in which object positions are computed based on the time it takes for electromagnetic or acoustic signals to travel from a small number of fixed transmitters (i.e. at least three for a 3D position solution) to a receiver embedded in each object. In contrast to electromagnetic positioning systems such as GPS, the TViews table is based on acoustic time-of-flight because acoustic wave propagation velocities are slower than electromagnetic wave propagation velocities, allowing high precision in the time of flight measurement without excessive electronics complexity or power consumption in the interaction objects. Furthermore, in the TViews system, the acoustic positioning signals are transmitted through the glass display surface itself and are passively monitored by the interaction objects. Since this acoustic positioning system does not require antenna grids or other materials covering the interactive surface, it is possible to provide an embedded digital display without suffering from display occlusion or other problems inherent in other technologies, regardless of the size or aspect ratio of the display surface. Furthermore, by broadcasting positioning signals from the display surface to all interaction objects simultaneously, there is no inherent limit to the number of interaction objects that may be used simultaneously on the same surface, and the position update rate is independent of the number of interaction objects. Our initial TViews prototypes have shown 100Hz update rates and millimeter scale position measurement accuracy, essentially invariant to the number of tracked objects.

In order to provide an **extensible object namespace**, TViews uses a globally unique digital identification number for each interactive object. The current 64-bit number allows an extremely large object namespace that is used in a hierarchical fashion to permit different object types to be differentiated through fields in the ID namespace. This ID space could be extended to a larger numbering space if necessary.

Supporting **portability** of interactive objects from one table to another requires two properties. First, a TViews table must be able to identify any object from the entire namespace as soon as it is placed on its interaction surface. Second, any TViews table, regardless of its size or aspect ratio, must be able to determine the position of any object that is placed on its interaction surface.

To support the first required property, an enumeration strategy is required for the singulation and identification of all objects that are currently on a given table even when the number of objects and their identities are not known a priori. Secondly, the position solution based on time-of-flight data is performed by the computational system within the table rather than on the objects themselves. This way, the objects do not need to know anything about the size of the table on which they have been placed, and only need to be able to measure acoustic times of flight and communicate them to the table.

In order to support an **extensible set of applications**, TViews provides an API (Application Programming Interface) layer based on an event model that sends events to TViews applications when objects are added, moved, manipulated, or removed from the table. Thus developers are able to easily create new applications for the table platform through the standard expedient of registering them for input events from the table's control system. TViews also provides an application and object manager to support multiple applications on the table and to keep track of interaction objects that are associated with different applications.

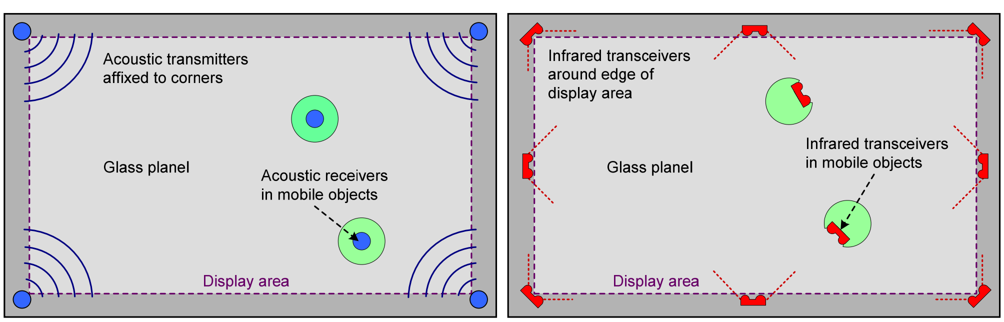

Figure 3:
Top view of the sensor layout on the surface of the TViews table.

TViews Hardware Implementation

As previously mentioned, TViews uses a combination of acoustic and infrared communication technologies to implement a table-based object positioning architecture conforming to the feature set outlined above. Inside the table, a master control board connected to the interaction surface itself manages the communication with and tracking of the large set of interaction objects (known as pucks) as they are placed and moved on its surface. Acoustic ranging pings are used to locate the pucks, while object identity, time of flight, and object manipulation information is communicated bidirectionally between the master control unit and all pucks by means of infrared transceivers.

To transmit acoustic ranging pings from the corners of the display surface, piezoceramic transducers are affixed to the bottom-side four corners of the display glass. These transducers launch ultrasonic acoustic waves at a frequency of 200Khz into the glass surface in a bulk longitudinal acoustic wave mode. Furthermore, a frame consisting of eight infrared transceivers is placed around the edge of the interaction surface for communication. A larger than required number of inexpensive infrared transceivers are placed around the edge of the display surface so that hand occlusion or object occlusion do not result in blocked communication between interaction objects and the TViews table. Each puck (interaction object) is equipped with an ultrasonic receiving sensor to pick up the acoustic waves as they travel through the glass display surface beneath, as well as an infrared transceiver for bidirectional data communication with the table. Figure 3 shows the layout of the sensors on the glass, and Figure 4 provides a diagram of the entire system.

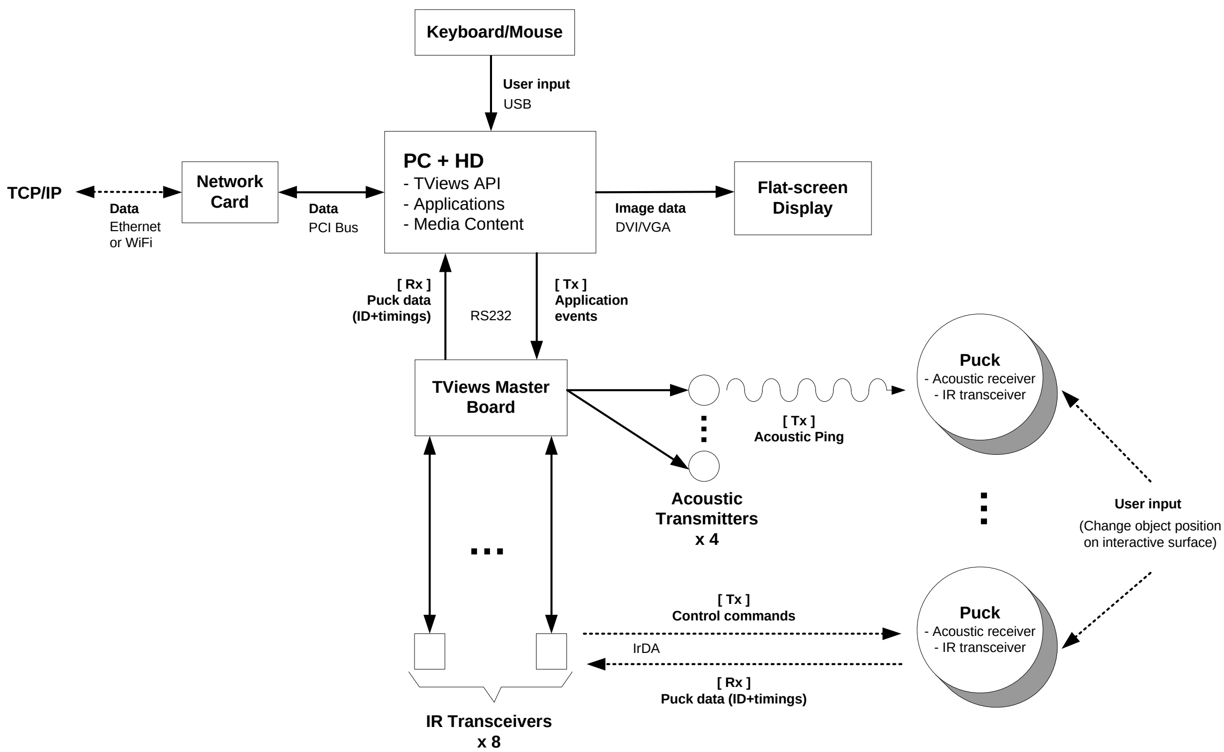

Figure 4: Diagram of the TViews system.

This position sensing approach was inspired in part by the Tap Window project, a system for tracking taps and other impacts on large glass surfaces [Paradiso 2005]. The Tap Window initially used pick-ups made of Polyvinylidene Fluoride (PVDF) piezoelectric foil, while later versions employed small piezoceramic microphones. The pick-ups were mounted to a large glass window and were used to track a range of different of taps, from low-frequency knocks generated by the impact of a fist to sharp taps generated by a metal object. An early version of the system was used to track the impact of ping pong balls on a table during game-play [Ishii 1999].

An important difference between the TViews sensing approach and previous work on acoustic tracking systems such as the Tap Window is that TViews simultaneously tracks many objects on the table rather than a one object at a time. Another important distinction is that the Tap Window is a passive system, used to track impacts on glass from inactive interaction objects such as metal objects or human fists, while the TViews table is used to track active objects containing sensing and computation at the level of the interaction object itself. Furthermore, the acoustic receivers in the TViews system are located in the interaction objects themselves, while acoustic transmitters are affixed to the corners of the glass. This reversal of the past approach endows the TViews display surface with a *positioning utility*, which enables the TViews system to scale to an essentially unlimited number of interaction objects on the table without a significant reduction in the refresh rate or accuracy of object positions.

The TViews Application Programming Interface

The application programming interface (API) for the TViews platform is implemented in Java and runs on the PC housed inside the media table. It allows developers to create different kinds of media table applications that make use of the combination of real-time object tracking and embedded display. The TViews software underlying the API keeps track of the incoming messages from the master control board and parses them into three main types of events: puck added, puck removed and puck metadata (e.g. position, button state, etc.) updated. The API employs a Java event model to fire notifications about these events to any applications that have registered themselves as listeners on the TViews platform. These events can be extended to incorporate additional event notifications to subscribing applications about user actions that make use of any external I/O devices that might be attached to a puck, such as a button press. The API will eventually be extended to include support for bidirectional messaging, which would allow the application send messages to the pucks or to control specific properties for each puck. For instance, a puck might flash to draw attention. Or if the puck is equipped with a small add-on display, the application might send a text, picture or video message. Future pucks might even contain actuators that allow them to move, vibrate, or provide haptic input or output.

TViews thus provides a scalable framework so many different media table applications can run on the same platform at the same time, as well as on many network-connected platforms. TViews interactive objects can be associated to a particular application based on their unique functionality or physical shape, or they can be used as generic controls for multiple applications. Moreover, the compact sensing and display design allows the table to be set up in everyday living environments where there is little or no support for infrastructure that is external to the table itself.

Tabletop Application Design

Over the past decade, tangible interface researchers have created prototype applications that range across a variety of application domains and contexts. Ullmer provides a good overview of application domains relevant to the broader area of tangible interface design and suggests that tangible interfaces are generally suited to activities that require or benefit from collocated cooperative work [Ullmer 2001]. Media tables in particular provide a shared horizontal workspace for these interactions, which is not necessarily the case for all tangible interfaces. Example application domains that have been explored for tabletop platforms include urban resource planning [Underkoffler 1999] and business management and supply chain visualization [Patten 2001].

Past work on tabletop applications and creating our own prototypes has inspired us to formulate a taxonomy of design considerations for media table applications, shown in Figure 5. This taxonomy highlights issues that are specific to the media table design space, and does not cover all aspects of the broader area of graphical user interface design as a whole. Nevertheless many of the issues that come up in graphical interface design are also encountered when designing for tabletops. For example, media content can be visualized on a display surface using approaches such as hierarchically layered views or continuous spaces that can be panned and zoomed. On shared tabletop displays, the design of the interactive objects, the approach used for interface and control, and the strategies for connecting applications across multiple tables come into play.

Considerations		Approaches	
		Less Constrained	**More Constrained**
Object Design	**Control and mappings**	Objects as generic handles or control devices (e.g. puck, stylus)	Objects with specific fixed meaning or functionality
	Extensions and customizations	Generic input/output add-ons (e.g. buttons, dials, sliders, lights, displays)	Physical shape reflects object's meaning in virtual space
Interface and Control	**Shared interaction approach**	Simultaneous movement of individual control points	Enforced turn-taking or simultaneous movement with coordinated control
	Multi-viewpoint approach	Directionless interface with free rotation of all virtual media objects	Fixed interface elements or automated re-orientation of media objects
Networked Tables	**Visual interface approach**	Tables provide independent views into the media space	Tables coordinate their views into the media space
	Remote object presence	Results of remote actions and object movements are displayed	All object movements and actions are displayed

Figure 5: Taxonomy of interface design considerations for the development of multi-user media table applications.

Certain types of applications might point to specialized solutions that impose constraints on interface and interaction design. Other types of applications might allow a greater level of flexibility in the interface and interaction design. We can therefore consider design solutions that range from less constrained to more tightly constrained approaches to interaction object control within each of these areas. In terms of interaction object choice, they might act as generic controllers, like a stylus that could be used to grab onto any virtual media object. Or they might be given a fixed meaning designated by their color, shape, or other physical attributes. For example in the Urp system [Underkoffler 1999], the interactive objects are permanently associated with a specific digital meaning and functionality based on their physical form, because Urp's physical building models are attached to corresponding virtual building models with specific physical forms; they therefore cannot be used to control a different kind of virtual media object without breaking the interface metaphor.

At the level of the visual interface, media tables are a shared platform so we must consider how multiple users interacting through multiple points of control will coordinate their activities, and how the design can be used to enhance the experience of all participants. This can be done through simultaneous free movement of objects, or with more constrained approaches like enforced turn-taking or coordinated control. For instance, user interaction with a picture sorting application might be best served by an interface that allows continuous and simultaneous movement of the interactive objects. In this way, multiple people could browse and organize a collection together, perhaps selecting images that interest them, and storing these images within their own particular interactive objects. In contrast, drawing a figure in a graphical drawing application might use coordinated control between two objects in order to anchor one corner and then stretch out the figure with a second object. This style of interaction was used in GraspDraw, a two-handed drawing application created on the ActiveDesk [Fitzmaurice 1995]. In the case of a digital board game, the rules of the game might impose a turn-taking approach in the game-play, which is common in many traditional tabletop games such as chess or checkers. However new types of games might also be designed to allow simultaneous collaborative engagement of players with multiple objects at once. Another important issue that arises in the visual design of tabletop applications is the question of image or interface orientation, since users are likely to view the display surface from different sides while seated around the table. Automatic orientation of interface elements on tabletop displays has been explored by the DiamondTouch researchers at MERL [Shen 2004].

Finally, tabletop designers need to think about how multiple networked media tables coordinate views and user interactions, which is in some ways parallel to the issues that arise when connecting desktop application environments. Multiple views into the media space can be tightly coordinated so that remote users will always see the same thing, while a less constrained approach might lead to each table acting as an independent window into a larger virtual environment. The latter approach is generally used in online role-playing games, where different players can be located in different parts of the virtual world at any given time, so each user sees a different view of the virtual world even though all users are interacting within the same virtual space.

Beyond the coordination of the visual interface, tabletop applications also need to consider how the presence of remote objects is displayed on connected tables. In a less constrained approach, connected tables might only see the results of actions that have been performed by remote objects, such as the placing of a game piece. Other applications might lend themselves to a tighter coupling of object presence. For example, a networked version of the Urp system displayed the footprints of remote buildings on a connected table so that remote users could see the way they were being moved in real-time [Underkoffler 1999].

Sample TViews Applications

To demonstrate the scalability and flexibility of the TViews system, two complete TViews media tables have been constructed and evaluated both separately and together as a networked system. One TViews table was located at the MIT Media Laboratory in Cambridge, MA, and the other was located around the world at Samsung's research center in Suwon, Korea. The two tables are shown in Figure 6. A total of eighteen interaction objects (called "pucks") and ten different sample media applications have been built over the course of the development and testing of the TViews table. All pucks and all applications can run on both tables, and the pucks can be swapped between applications running on different tables. Several of the applications make use of the expandable I/O feature of the pucks through the use of add-on buttons that provide customizable functionality within the application space. We will now describe several of these TViews applications and explain how these different applications coexist on the TViews platform.

Figure 6:

TViews media tables were constructed in two different furniture styles at the MIT Media Lab (left) and Samsung Research (right).

Picture Sorter

Over the past decade, we have moved into the era of digital photography. Rolls of film are quickly being replaced with ever-increasing capacities of digital storage, and the lengthy process of analog processing that generates viewable photographs from rolls of film negatives has been replaced with the instant gratification of immediately viewable images. These photos can be shared via mobile messaging, downloaded to laptop and desktop PCs, uploaded to picture sharing websites online, or simply printed out in regular paper form. This transformation in the photographic process allows people to quickly capture far greater numbers of images, but it also requires better tools for organizing these large image collections.

Many such tools exist, such as Apple's iPhoto or Google's Picasa, but managing large image collections can be a tedious and repetitive task as it often involves manually entering keywords such as file or directory names on a computer. This is usually a solitary activity that does not facilitate face-to-face interactions between people. Thus the leisurely group activity of sorting through a pile of physical photographs on a table has been replaced with single-user point-and-click interfaces for image organization. The TViews picture sorting application explores how a tabletop platform can bring back some of the shared fun into the process of organizing digital photographs. The application makes use of the metaphor of a shoebox of images that is dumped onto a table surface for sorting. New images appear in a pile at the center of the TViews table, and the pucks are used to sort them into smaller clusters. Unlike physical photographs which can only be in one pile at a time, the digital nature of the application allows linked copies of the same photograph to be sorted into several different clusters. While the application currently provides only basic sorting functionality, users would eventually be able to provide annotations for their photographs and the system could incorporate intelligent sorting features based on these annotations or on other aspects of the images. The image clusters could also be saved as slideshows, posted to the web for remote access, or viewed within different browsing views such as the map browser discussed in the following section. Figure 7 shows the picture sorting application running on the TViews table.

Figure 7:

A collection of images can be organized in the Picture Sorter by dragging images into small clusters around the surface of the TViews table, similar to the way physical photographs are sorted into piles on a tabletop.

Map Browser

As digital photography continues to spread and evolve, camera technologies become more intelligent, and the information they capture along with the visual image itself allows software tools to automatically classify and retrieve images based on their metadata tags. Nowadays, most digital cameras store the date and time as well as a variety of additional information, such as the aperture, shutter speed and other camera settings that were used when the image was taken. Certain cameras, including many of the camera-equipped cell phones, allow users to add voice annotations to their photographs. It is also possible to attach GPS (Global Positioning System) receivers to some digital cameras in order to record geographic location information. One can imagine that in the not-so-distant future digital cameras will also incorporate sensors to keep track of ambient or environmental information such as weather conditions. The TViews map browser takes advantage of GPS metadata to organize images on a spatial map based on the time and location at which each picture was taken.

In the Map Browser, a timeline on one side of the table is color coded by day and small colored dots appear at each photograph location on the map, indicating the days that photographs were taken there. Users attach the pucks to different days on the timeline and drag them around the map to reveal that day's images. The images appear clustered around the puck, and can be zoomed by pressing the button on top of the puck. When an image is zoomed, another puck can be used to grab hold of it and drag it to another part of the table for separate viewing. Figure 8 shows the map browser displaying a collection of images from a vacation on the west coast of Ireland.

Figure 8:

In the Map Browser, the pucks are used to navigate a collection of images on a geographic map. In this case, the image collection is from a vacation on the west coast of Ireland.

Pente

Games of many different kinds play an important role in human recreational and social activities. Games can be both competitive and cooperative in nature. They can make use of skill, strategy or chance, and they can be played in many forms, with cards, boards, tiles or even the entire body. With the advent of digital game technology, game designers have made extensive use of emerging digital technologies to enhance game-play by immersing viewers into virtual worlds with stunning graphics and complicated artificial intelligence (AI) based rule engines. While many of these games can be played in a networked mode with thousands of players at a time, they no longer provide the face-to-face social interactions common to more traditional games.

We have implemented a simple Pente board game for the TViews table to demonstrate digital board game play on the TViews platform. The game can be played with two or three players, and the goal is to place five stones in a row on the grid or to capture five pairs of an opponent's stones. Each player receives a puck, allowing them to drop stones onto the grid. There are three different pucks used for the game: yellow, red and blue. The yellow puck comes with a yellow removable plastic icon-cap on top, and it can place only yellow stones on the table. This behavior demonstrates how the pucks can be physically customized for a particular application, and how the physical shape of the interactive objects can be permanently linked to different meanings or functionality within the virtual space of the application.

Moreover, if the red puck is moved to a different TViews table running the Pente application, it will retain its identity as a red puck in the game. Figure 9 shows the game of Pente on the TViews table. In addition to the basic game features, Pente can be played in a networked mode from two different tables at once. In this case, each player's moves show up on both tables at the same time. For instance, two players in the game might be located at one table, while the third player is located at a different physically remote table. The idea of networked tabletop game-play could eventually be extended to include many existing networked games, such as online role-playing or simulation games.

Figure 9:

This game of Pente on the TViews table supports two or three players at a time and can also be played in networked mode between multiple tables.

Springlets

Computers enable a variety of improvisational interactions with visual or auditory media, ranging from simple painting programs to real-time control of music or video sequences in a media performance. The TViews system provides a shared space for these kinds of interactions to unfold.

The Springlets application is a simple example of visual improvisation on the TViews table. Two virtual spring objects (modeled as masses connected together with springs) can be controlled by the pucks. The spring objects leave colorful trails behind them as they bounce around the display area. Users latch onto the masses with a button press, and drag them around the table causing the attached masses to follow behind. A second button press drops the masses, propelling them forward on their own. Once the spring objects are in motion, they dance around the table and users can engage in improvisational play as they try to trap the masses in order to control the movement and display of the colorful trails on the display area. Figure 10 shows the Springlets application running on a TViews table.

Figure 10:

In the Springlets application, the pucks are used to propel masses around the table that leave colorful trails behind them as they bounce around the screen.

Conclusion

Most prior media table prototypes have been one-shot approaches which were used to demonstrate the value of specific interaction or application concepts. The community has not yet focused on creating a strong underlying technical framework that will provide the winning combination of economic viability and richness of application possibilities that will be required before media tables can become as ubiquitous as desktop GUIs are today. This has been historically true because the technology of interactive tables has remained unreliable and cumbersome, preventing media tables from making the leap into our everyday environments and gaining widespread use.

In this work we present a general purpose model for media table platforms that allows large numbers of interactive objects to be built and to run across multiple platform instances at once, with an economical and scalable sensing mechanism. Now that we have an extensible method for creating media tables and a straightforward API for tabletop development, we would like to grow the community of developers who can create applications for the platform. Many of these applications can be based on ideas that our colleagues have already described. For instance, the ability of TViews tables to be networked together opens up the application space for new kinds of multi-person game play that can bridge the worlds of pre-existing physical and digital games.

Together with the development of the application space, we would also like to push forward the design of different interaction objects, providing custom shapes for different applications, as well as additional input/output elements like small displays or sensors within certain interaction objects. We would also like to incorporate interaction object capabilities into existing digital media devices, such as digital audio players or digital cameras, in order to provide new ways of sharing and exchanging the digital contents within them as they are placed on a TViews surface. If we continue to grow the media table application space, we hope this will feed back into increased investment into the development of the platform itself. Through this cycle of growth that pushes on both the platform and application development in turn, we hope that media tables will fulfill their promise and will one day be part of our everyday environment.

References

[Dietz 2001] Dietz, P., Leigh, D. (2001) "DiamondTouch: A Multi-User Touch Technology" in *Proceedings of the 14th Annual Symposium on User Interface Software and Technology (UIST '01)*, ACM Press, pp.219-226.

[Fitzmaurice 1995] Fitzmaurice, G., Ishii, H., Buxton, W. (1995) "Bricks: Laying the Foundations for Graspable User Interfaces" in *Proceedings of Conference on Human Factors in Computing Systems (CHI '95)*, ACM Press, pp 442-449.

[Ishii 1999] Ishii, H., Wisneski, C., Orbanes, J., Chun, B., Paradiso, J. (1999) "PingPongPlus: Design of an Athletic-Tangible Interface for Computer-Supported Cooperative Play" in *Proceedings of Conference on Human Factors in Computing Systems* (CHI '99), ACM Press, pp. 394-401.

[Mazalek 2003] Mazalek, A., Davenport, G. (2003) "A Tangible Platform for Documenting Experiences and Sharing Multimedia Stories" in *Proceedings of ACM SIGMM 2003 Workshop on ExperientialTelepresence (ETP '03)*, ACM Press.

[Paradiso 2005] Paradiso, J.A, Leo, C.-K. (2005) "Tracking and Characterizing Knocks Atop Large Interactive Displays" in *Sensor Review*, Vol. 25, No. 2, pp.134-143.

[Patten 2001] Patten, J., Ishii, H., Hines, J., Pangaro, G. (2001) "Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces" in *Proceedings of Conference on Human Factors in Computing Systems (CHI '01)*, ACM Press, pp.253-260.

[Shen 2004] Shen, C., Vernier, F.D., Forlines, C., Ringel, M. (2004) "DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*, ACM Press, pp.167-174.

[Ullmer 1997] Ullmer, B., Ishii, H. (1997) "The metaDESK: Models and Prototypes for Tangible User Interfaces" in *Proceedings of Symposium on User Interface Software and Technology UIST '97*, ACM Press, pp.223-232.

[Ullmer 2001] Ullmer B., Ishii, H. (2001) "Emerging Frameworks for Tangible User Interfaces" in *Human-Computer Interaction in the New Millenium*, Ed. John M. Carroll, Addison-Wesley, 2001, pp.579-601.

[Underkoffler 1999] Underkoffler, J., Ishii, H. (1999) "Urp: A Luminous-Tangible Workbench for Urban Planning and Design" in *Proceedings of Conference on Human Factors in Computing Systems (CHI '99)*, ACM Press, pp.386-393.

[Wellner 1993] Wellner, Pierre (1993). "Interacting with Paper on the Digital Desk" in *Communications of the ACM*, July 1993, Vol. 36, No. 7, pp.87-96.