Pen-Based Computing

Pens may seem old-fashioned, but some researchers think they are the future of interaction. Can they teach this old dog some new tricks?

By Gordon Kurtenbach

DOI: 10.1145/1764848.1764854

When I entered graduate school in 1986, I remember reading about the idea of using a pen as an input device to a computer. Little did I know that the idea had been around for long time, from the very early days of modern computing. Visionaries like Vannevar Bush in his famous 1945 article "As We May Think" and Ivan Sutherland's SketchPad system from the early 1960s saw the potential of adapting the flexibility of writing and drawing on paper to computers.

The heart of this vision was that the pen would remove the requirement for typing skills in order to operate a computer. Instead of typing, a user would simply write or draw, and the computer would recognize and act upon this input. The rationale was that by supporting this "natural" expression, computing would be accessible to everyone, usable in broad range of tasks from grandmothers entering recipes, to mathematicians solving problems with the aid of a computer.

Like many visions of the future, this one was inspiring but not wholly accurate. Certainly some of the key technologies to enable pen-based computing have come into fruition and have been adopted widely. However, the dream of ubiquitous handwriting and drawing recognition has not materialized. One can argue that this type of technology has yet to mature but will in the future.

What's fascinating about pen-based computing is how it is being used in alternative ways from the original vision, which was only a slice of the rich variety of ways a pen can be used in human-computer interaction. This article is about those other things: the ways in which pen input to a computer has been found to be valuable, along with where it is going.

"The original vision of pen-based computers was that they would bring the benefits of physical paper and pen to computer interaction... allowing people to interact more 'naturally' with the computer instead of typing."

PRACTICALITIES

There are some very practical issues that have dramatically affected the adoption of pen-based systems in the marketplace. Earlier work on computer input techniques, coming from a heritage of data entry, largely abstracted away some of the practical differences to present a more programmatic or "data centric" view of computer input devices.

Early work on interactive computer graphics by Foley and Wallace classified mouse and pen input as pretty much the same thing: both provide an x and y location and are capable of signaling an event (a pen press or mouse button press). However, later, researchers (such as Buxton) documented many subtle but important differences that affect the suitability of an input device for a specific task. For example, the mouse has some very practical properties that make it a successful and ubiquitous input device for desktop computers. It is a very efficient pointing device and allows the cursor location to remain unchanged when buttons are clicked.

Similarly, the pen has a set of very practical properties that define the contexts in which it will be effective. For example, one annoying aspect of pens is that they can be misplaced or lost, a problem that is exacerbated in the mobile device context. But it can be overcome by tethering the pen to the computer, or alternatively, the computer industry has recognized in many situations pointing by touch, without a pen, is sufficient.

The reverse has been used to an advantage too. For example, electronic white boards like Smart Board use multiple pens as an easy way to switch between ink colors when drawing.

Another practical but subtle and vexing issue with pens is that they require picking up. What happens when users want to switch hands or switch which input device they're holding? Some users become adept at keeping the pen in their hand while typing or using the mouse, but that is a generally inefficient and inelegant solution.

The key observation is that there is a rich set of issues and preferences surrounding any particular computer input situation and, in many of these cases, even if perfect handwriting and drawing recognition were available, the pen would still not be a preferred choice. Pen input is not an effective input technique if you cannot find the pen, or if one can simply type faster than write or draw.

THE ART OF SKETCHING

When is the pen a good choice? One task where the pen has a fundamental advantage is drawing. But even then, one has to be very careful to identify the precise drawing task.

The original vision of the user drawing diagrams and having the computer recognize and replace the rough drawings with formal structured graphics is not what the pen does best. Historically, the Achilles' heel has been getting the computer to recognize properly what the user has drawn. However, if the goal is to create formal structured drawings, then why not create formal

Figure 1: (a) Pen input is essential to the art of sketching. Pen-based tools like SketchBook Pro used with a Wacom tablet make powerful free-form sketching and painting tools that capture the manual skills of an artist.

(b) The same activity is still compelling in small formats (like the iPhone), even drawing with the finger. (c) The quality of drawing that can be performed with SketchBook Mobile for the iPhone is very high.

Figure 2: The user can interact with marking menus in two ways. (a) Selection can be made by popping up menus; or (b) once the user has memorized a path through the menus, selection be made much faster with a quick mark representing the path.

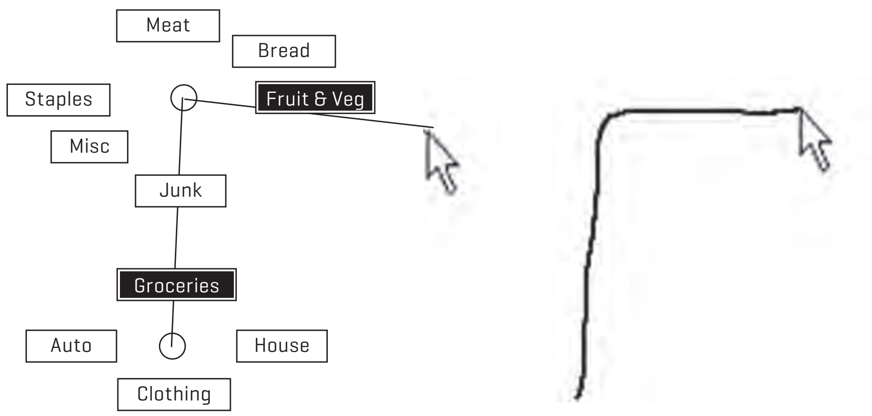


Figure 3: Marking menus use the spatial mapping between a vocabulary of zigzag marks and a hierarchy of radial menus. This permits easy to draw marks to be associated to arbitrary menu commands.

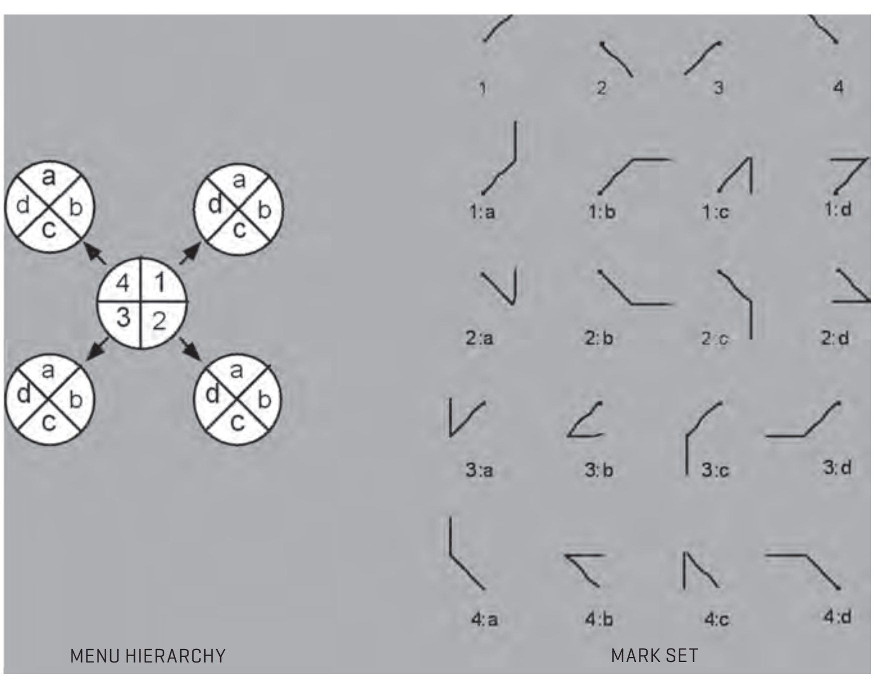


Figure 4: The Unistrokes character entry technique reduces letter input to single stroke marks that are easy to draw and easy to recognize.

structures directly? Unfortunately for the pen, mouse-based computer interfaces for drawing structured graphics are effective and arguably have become a more "natural" way to create them. Nowadays reverting from mouse to pen would displace industry practices.

However, if you need hand-drawn graphics, like the kind you make when drawing with a pen on paper, pen-based input shines. While this may appear to be trivial example, its value is constantly misunderstood and underestimated. The appeal of sketching never fails to amaze me. I was a member of the original development team that created the first version of Autodesk SketchBook Pro, a paint application designed specifically for sketching and capturing hand-drawn ideas with all the sensitivity of drawing with pencils, pens, and markers on paper (see **Figure 1**). Recently, I was approached with the idea of making a version of SketchBook for the Apple iPhone. While I thought it was a cool technical challenge, I could not imagine a really good reason why someone would want to use it.

Fast-forward several months and the more fully featured paint program SketchBook Mobile went on sale on the iTunes App Store. Its popularity was a pleasant surprise and it became one of the most downloaded applications on the store. What was even more amazing was what people were drawing with it. **Figure 1b** and **1c** show an example. Obviously, the artists creating these works were enamored with sketching even on this small format, or rather, they were interested in sketching because it was this format – the iPhone.

PEN-BASED INPUT≠EASE OF USE

While the pen allows high-quality sketching, it also sets the user's expectation for symbol recognition. Give pens to users, and they expect to be able to input handwriting and symbols and have the computer recognize them. I learned this lesson years ago demonstrating a little program that allowed the user to create circles, squares, and triangles by drawing them with a pen. I was demonstrating the program as part of our laboratory's "open house" to the general public. People came by my booth and I explained how you could use a pen to input commands to the computer and showed examples drawing circles, squares, and triangles where the computer replaced the hand-drawn objects with "perfect" ones. I then asked people to try it themselves. Surprisingly, most people didn't try drawing circles, squares, or triangles but tried writing their names. People's expectations were that anything could be recognized and this overrode any of my instructions beforehand. People expect a system with very general symbol recognition the moment a pen reaches their hands.

The subtle lesson here is that pen input or symbolic input is not inherently "easy to use" because it does not reveal to the user the capabilities of the system. This is a critical insight and it is this property of "self-revelation" that makes modern graphic user interfaces "easy to use"—specifically, by displaying graphics like text, icons, pictures, and menus, the computer "reveals" to users what they can and can't do, where, and when. We can think of graphical interaction widgets like buttons, and menus as "self-revealing." The method for finding out what functions are available and invoking those functions are combined into the same entity, namely, the button or menu.

Symbolic markings made with a pen are not self-revealing. A user can draw any shape or symbol and there is nothing intrinsic in that interaction that shows the user what marks the system recognizes or what they do. One simple method of "revealing marks" is to display a "cheat sheet," a list that shows the correspondence between a mark and the command it invokes. For example, a cheat sheet might show that drawing "C" copies and drawing "V" pastes.

Much research on pen-based user interfaces involves systems in which symbolic marks are used. Early work by C.G. Wolf used the symbolic language from paper and pencil proofreaders editing marks to design a system to edit text using a pen to mark up a document. Similarly, a system called MathPaper supports inputting mathematical formulas using the pen. This approach holds the promise of commands being easy to remember and perform. It's easy to remember "C" is for copy and it is also very quick to draw the "C" with the pen. However, complications arise when many symbolic marks are needed, and there is no prior existing vocabulary of marks. If more than a dozen "command marks" are needed it becomes difficult to design meaningful ones. In this case, it would be more effective to use existing graphical user interface techniques of icons, button, or menus for these commands.

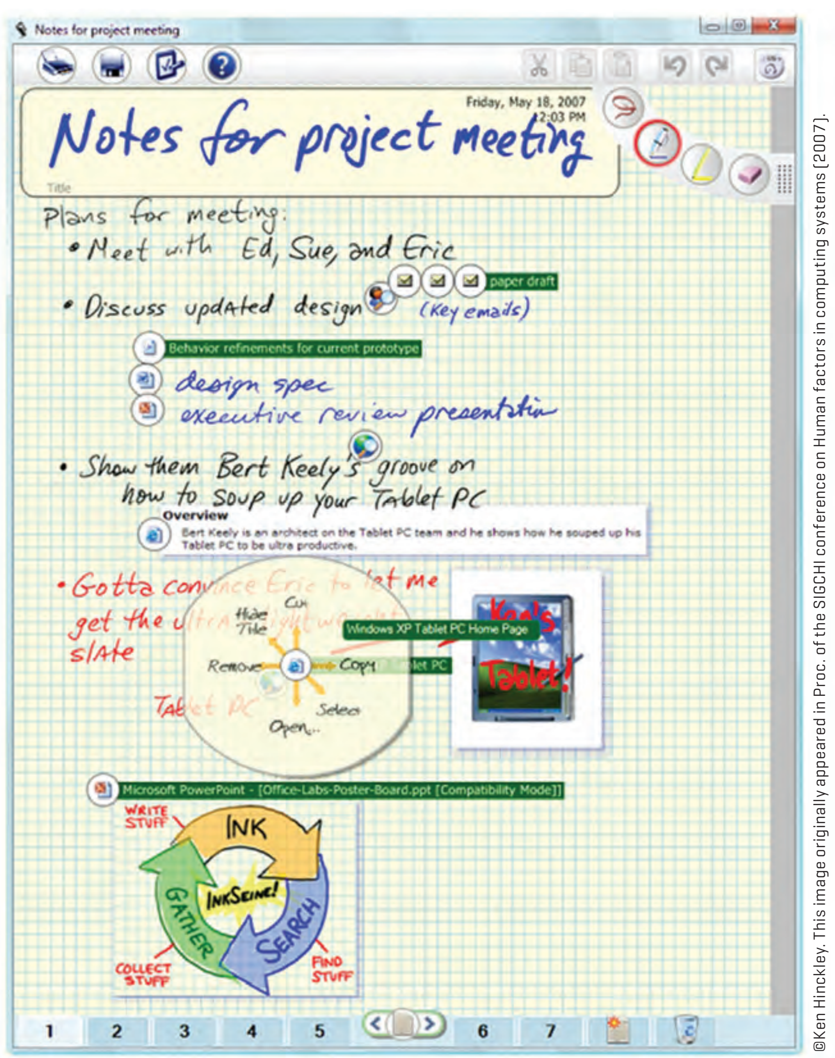
MARKING MENUS

A technique called marking menus was developed to address this problem. Marking menus combine a vocabulary of marks with a pop-up graphical menu to allow a user to learn and use marks that are easy and fast to draw. Like icons and buttons, marking menus interactively reveal the available functionality, and the computer recognition of the marks is simple and reliable. Essentially, marking menus combine the act of revealing functions to the user and drawing a mark.

A simple marking menu works as follows: As with a regular pop-up menu, a user presses the pen down on screen and a menu pops up. This menu differs from the linear menus we commonly see in that the menu items are displayed radially in a circle surrounding the tip of pen. A user can select a menu item by moving in the direction of the desired item and lifting the pen (**Figure 2**).

Marking menus are designed so

Figure 5: InkSeine combines the advantages of the free-flow of a pen and paper notebook with direct manipulation of digital media objects.



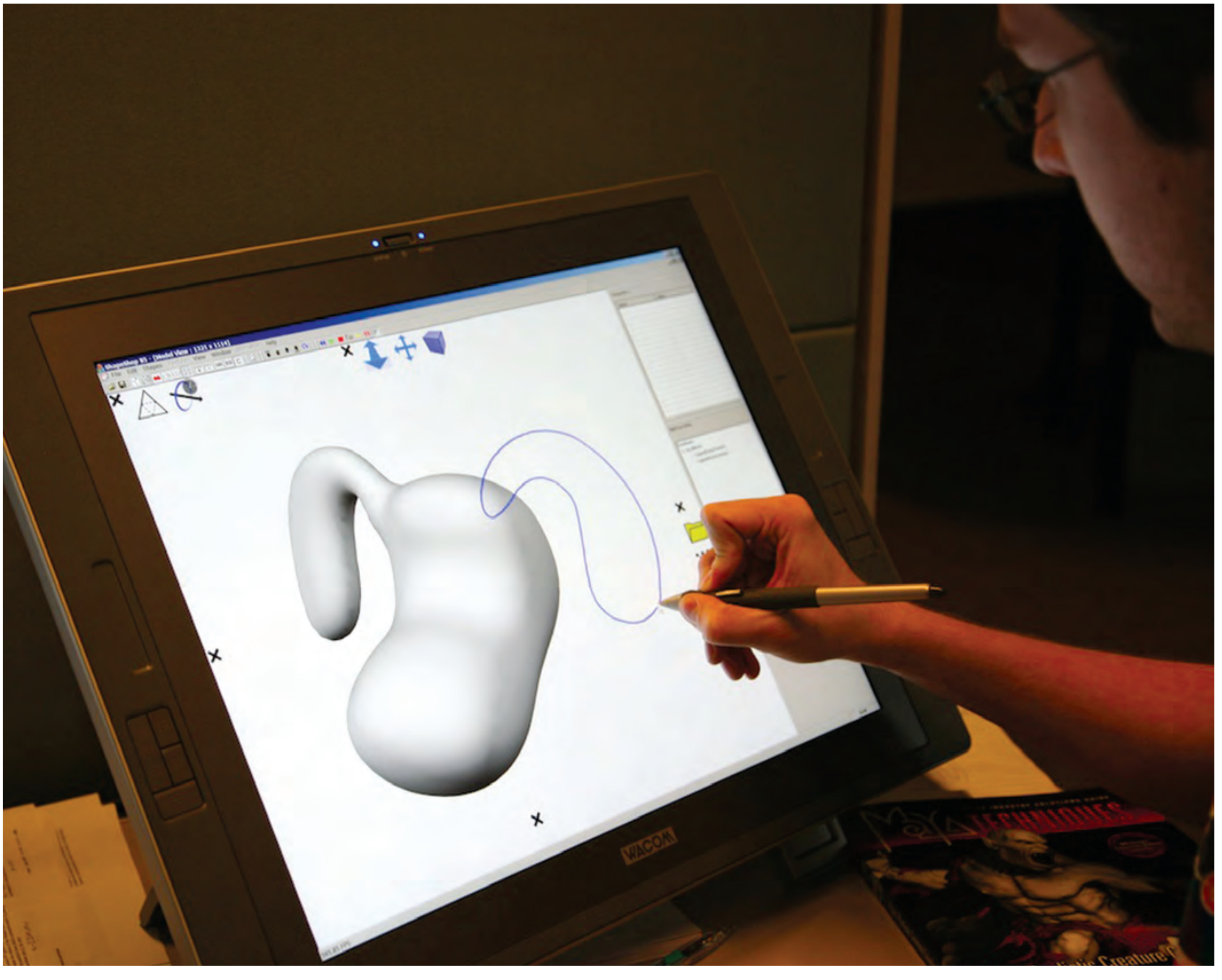
that a user does not have to do anything special to switch between selecting from the menu and using a mark. If the user presses the pen down and hesitates (waiting for the system to show what's available) the menu is displayed and selection can be performed in the usual way by pointing to menu items. However, if the user presses the pen down but does not hesitate and begins to move right away, a mark is drawn. This way, a user can gradually move from selecting a command via the menu to selecting by drawing a mark. Novices to the system can pop up the menu to recall the location of a particular menu items. With practice, the user memorizes the location of menu items and can select them quickly by making a quick mark in that direction. Novice use is a rehearsal of expert performance. Research has shown that selection with a mark can be up to ten times faster than popping up the menu, making this technique very useful for frequently selected menu items.

Marking menus takes a meaningless vocabulary of zigzag marks and makes correspondence to hierarchical menu items (see **Figure 3**). The result is a technique for providing an easy-to-learn, easy-to-draw, shorthand symbol set to access arbitrary computer functions. Much research has been conducted on marking menus, and they have been successfully deployed in commercial CAD applications.

Other similar clever ways of exploiting pen input beyond recognizing traditional handwriting and symbols have been explored. The text entry systems Graffitti and Unistrokes resulted from research that analyzed what types of symbols are easy for a user to draw and easy for the computer to recognize, in hopes of supporting handwriting input that is easier, faster, and more reliable than traditional handwriting. **Figure 4** shows how Unistrokes redesigned the alphabet to support this.

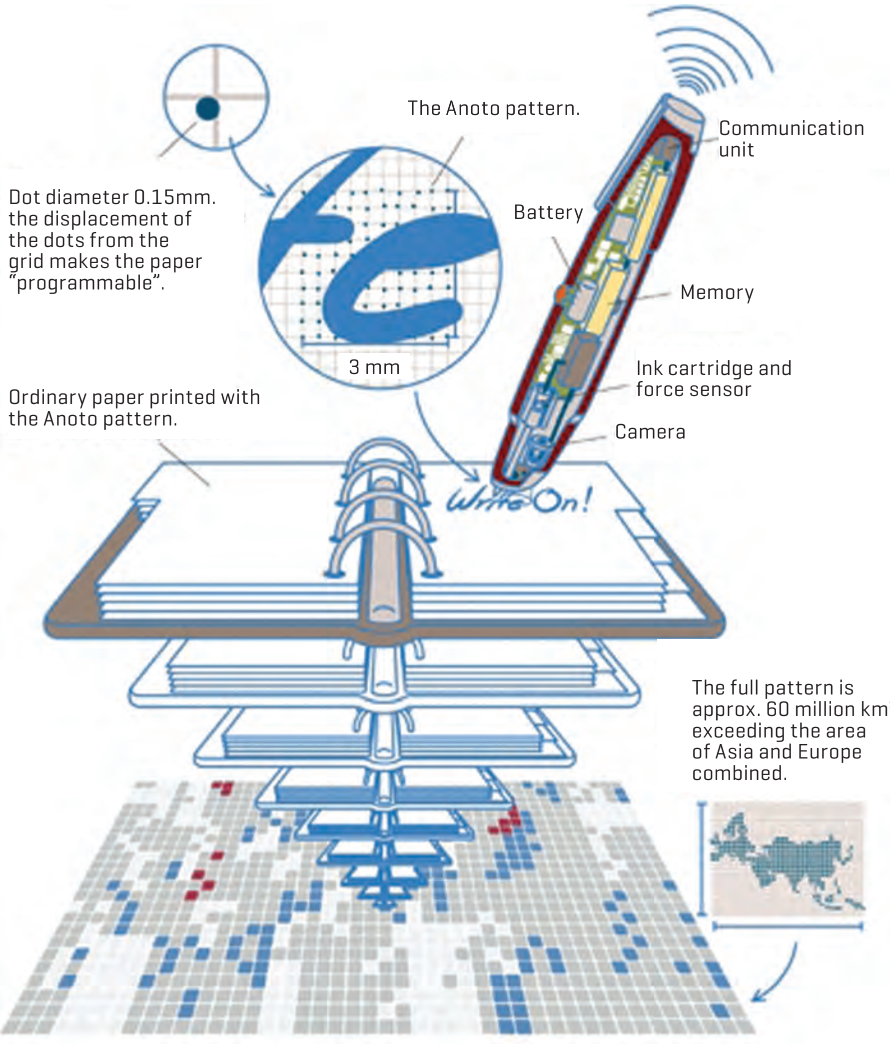
The concept behind marking menus has also been applied to handwriting input. Shumin Zhai and other researchers at the IBM Almaden Research Center

Figure 6: The application ShapeShop allows 3D shapes to be created quickly by inputting strokes and converting them to 3D graphics. Here the outline of the dog's ear is drawn and will subsequently be translated into a "blob" that corresponds to the shape of the input stroke and then connected to the dog's head.



developed the SHARK system, a graphical keyboard on which the user can input words by dragging from key to key with the pen. The path being dragged essentially creates a symbol that represents a particular word. Experiments on SHARK have showed that the user can learn this method of input and become proficient with it, and with practice, the rate of input can match touch-typing rates. As with marking menus, SHARK is a compelling example of how researchers are endeavoring to exploit a human's skill with the pen and ability to learn, to create human computer interactions that go beyond emulating traditional paper and pen.

Figure 7: A small camera in the tip of an Anoto pen allows the pen to sense and record what has being written and where.



PEN, THE GREAT NOTE-TAKER

Pen input research has also focused on the pen's ability to be used to fluidly switch between text input, drawing and pointing. Inspired by how people combine both drawing and handwriting in paper notebooks, these types of systems attempt to recreate and amplify this experience in the digital world. Ken Hinckley's InkSeine application is a prime example. With InkSeine a user can quickly throw together notes where ink, clippings, links to documents and web pages, and queries persist in-place. **Figure 5** shows an example. Unlike a mouse and keyboard system, where a user must switch between keyboard and mouse for text entry and pointing, InkSeine supports all these operations in a free form way, reminiscent of paper notebooks. (InkSeine is available as a free download from http://research.microsoft.com/en-us/redmond/projects/inkseine/.)

PEN INPUT 3D

Pen input can also be used to create 3D graphics. The goal of this work is for a user to be able to draw a perspective view of an object or scene and have the computer automatically recognize the 3D shape and recreate it accurately. A simple example is drawing a box in perspective and the computer automatically recognizes it as cube structure and creates the corresponding 3D geometry of a cube. The user can then rotate the object to see it from different viewpoints.

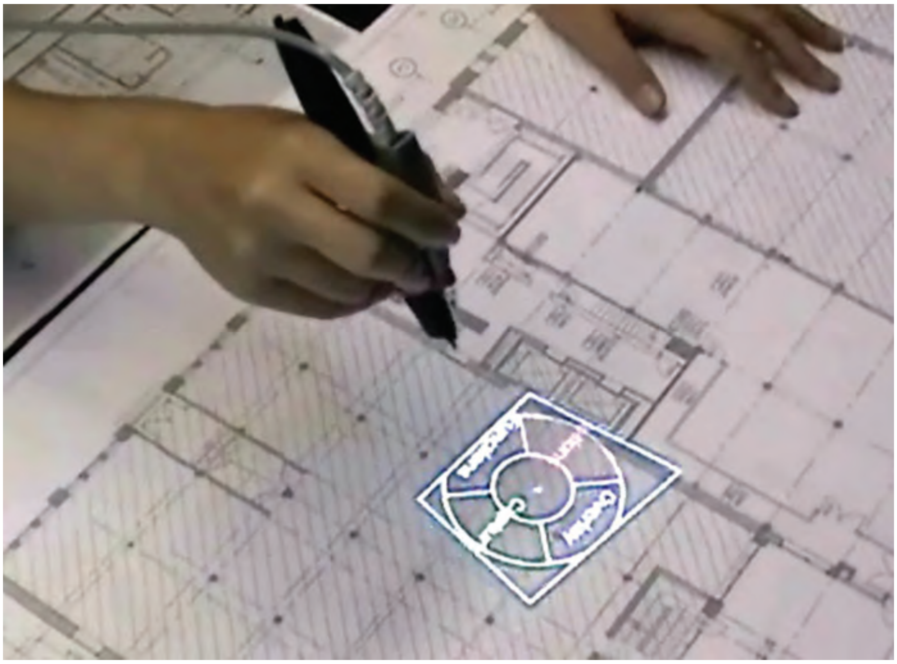
In general, this is a computer vision problem—recognizing shapes and objects in the real world—and much progress on this general problem remains. However, researchers have made progress with systems where pen input is interpreted stroke by stroke into three-dimensional objects. Research systems such as ILoveSketch and ShapeShop (shown in **Figure 6**) allow drawing on planes in 3D space or directly on to 3D surfaces.

BEYOND

Recently multi-touch systems, where the computer screen can sense multiple finger touches simultaneously, became a very hot topic for commercial usage. Some multi-touch systems are capable of sensing touch and pen

"Researchers and designers have made the key observation that the pen interactions are distinctly different from touch interactions and can be exploited in different and interesting ways."

Figure 8: The PenLight research prototype: A digital pen senses its location on a building blueprint and a micro projector mounted on the pen displays an overlay of information accurately positioned on the blueprint.



input. Work by Balakrishnan has shown systems where the dominant hand holds the pen and non-dominant hand controls the frame of reference (e.g., rotation of the image being drawn on) are effective and desirable ways of interacting. Work by Hinckley has shown the benefits of a simple design in these multi-touch and pen systems where "the pen writes," "touch manipulates," and "the two combine for special functions."

Ultimately, how to use the pen in combination with multi-touch will be largely determined by needs of applications. However, researchers and designers have made the key observation that the pen interactions are distinctly different from touch interactions and can be exploited in different and interesting ways.

Research has looked at ways of augmenting the pen with special hardware functions to create "super pens." The Anoto pen is a digital pen with a tiny camera embedded in its tip (**Figure** 7). The pen can store handwriting, markings, and pen movements, and identify on which document they were made and precisely where the markings are. This allows the world of paper and digital technologies to be combined.

This concept has been adapted from paper to physical 3D objects. Guimbiere and Song developed a computer-aided design system where a user could make editing marks on 3D physical objects using an Anoto pen. The objects, printed with a 3D rapid prototyping printer, have the same type of invisible markings as the paper allowing the pen to identify the object and the location of markings. Drawing a doorway on the side of physical model of the building causes the doorway to be added to the virtual model.

Pushing these ideas further, research has explored augmenting these "super pens" with output devices, like a LED screen that displays a line of text and an audio speaker. The PenLight system pushes this even further and uses a micro projector mounted on the pen that allows it to act like an "information flashlight." With the PenLight, the camera in the pen not only knows the location of the pen relative to the document it is over, but with the micro projector it can overlay relevant information. **Figure 8** shows the Pen Light system being used over a blueprint for a building.

"Some super pens, like Anoto's, can store handwriting, markings, and pen movements, and identify on which document they were made and precisely where the markings are. This allows the world of paper and digital technologies to be combined."

THE PEN TO COME

The original vision of pen-based computers was that they would bring the benefits of physical paper and pen to computer interaction by utilizing handwriting input and free-form drawing, allowing people to interact more "naturally" with the computer instead of typing. However, because reliable handwriting recognition wasn't available and people needed typeface text for a majority of tasks, the keyboard and mouse became more popular—so much so that the keyboard and mouse now seem to be the "natural" means of interacting with a computer.

But the pen remains essential for some tasks, like sketching and free form idea input, and in these applications it has found success. Furthermore, researchers explored ways of using pen input beyond emulating pen and paper, such as marking menus and PenLight, and this research has resulted in useful, successful and interesting technologies. Using this type of thinking, many interesting and inspiring explorations for pen input remain ahead.

Biography

Dr. Gordon Kurtenbach is the director of research at Autodesk Inc., where he oversees the Autodesk Research group whose focus is on human-computer interaction, graphics and simulation, environment and ergonomics, and high-performance computing. He has published numerous research papers and holds more than 40 patents in the field of human-computer interaction.