The IrPen

A 6-DOF Pen for Interaction with Tablet Computers

Jaehyun Han and Seongkook Heo ▪ *Korea Advanced Institute of Science and Technology*

Hyong-Euk Lee ▪ *Samsung Advanced Institute of Technology*

Geehyuk Lee ▪ *Korea Advanced Institute of Science and Technology*

Besides incorporating the IrCube tracker's basic operating principles, the IrPen takes into account tablet-specific requirements. It employs a sensor small enough for tablets, and the pen structure minimizes issues caused by reflections from a tablet surface. The design also addresses issues related to occlusion and ambient lighting.

Traditional pen interfaces use the pen tip's 2D position on a surface. To increase the input vocabulary, researchers have also used the pen's orientation and distance from the surface. For example, Tovi Grossman and his colleagues[1] and Sriram Subramanian and his colleagues[2] presented systems that use the pen's 3D position over a surface. Xiaojun Bi and his colleagues employed pen orientation to call a context menu.[3]

Furthermore, researchers have used these input dimensions to make 3D manipulation more intuitive. For example, a ray-casting method used a pen's position and orientation to select objects in a 3D virtual environment.[4] Bill Baxter and his colleagues used a pen's six degrees of freedom (DOF) to implement a model of the physical characteristics of various types of brushes for drawing applications.[5]

However, current pen systems have a limited tracking range and therefore don't fully support these interaction techniques. So, researchers have used alternative sensing tools such as the Vicon motion-tracking system or a 9-DOF sensor module.[3,6] But these tools aren't practical for tablet computers. (For more on pen interfaces, see the sidebar.)

To create a practical 6-DOF pen system for tablets, we reviewed 6-DOF tracking methods, including magnetic, optical, and acoustic trackers. We found that the IrCube system[7] had the most potential because its hardware structure comprises a few photo sensors and LEDs that we could easily embed in a tablet and pen, respectively. So, we extended the IrCube to create the IrPen system (see Figure 1). We defined additional requirements for a 6-DOF pen system specifically for tablets. On the basis of these requirements, we implemented a prototype and demonstrated its feasibility by measuring its performance.

The IrCube

The IrCube tracks the position and orientation of a pointer comprising an LED array, with each LED at a different orientation. Sensors measure the LEDs' intensities; then the IrCube uses an optimization algorithm to solve an inverse problem. When a sensor is stationary and an LED is moving, the intensity measured at the sensor changes with the LED's position and orientation. This forward model is

$$F = F_0 \times \frac{1}{d^2} \times D_{\mathrm{LED}}\left(\theta_m\right) \times D_{\mathrm{diode}}\left(\phi_n\right),$$

where F_0 is the intensity of the mth LED, d is the distance between the LED and the nth sensor, D_{LED} and D_{diode} are the LED's and diode's directivities, and θ_n and ϕ_m are the angles between a sensor and an LED relative to each other. Figure 2 diagrams the model.

The LEDs turn on and off sequentially, one at a time, and the IrCube measures their light intensity at each sensor. We measure the $m \times n$ sensor values for a complete illumination cycle. We can use the forward model to predict these values if we know each sensor's position, orientation, directivity, and sensitivity, and each LED's position, ori-

 Published by the IEEE Computer Society 0272-1716/14/$31.00 © 2014 IEEE

entation, directivity, and intensity. To determine the pointer's position and orientation, we use an optimization algorithm that minimizes the error function E:

$$E(x,q)=\sum_{m=1}^{M}\sum_{n=1}^{N}\left(O_{mn}-F\left(x_m-x_n,q_m\right)\right)^2,$$

where O_{mn} is the sensor measurement, $F(x_m - x_n, q_m)$ is the prediction of the forward model, x_n and x_m are the position of the nth sensor and the mth LED, and q_m is a quaternion vector representing the pointer's orientation.

The IrCube's large sensors (145 × 82 mm) make it problematic for tablets. In a demonstration of its application to tablets, the sensors were as large as the tablet.[7] The reported position error was 23.5, 20.2, and 27.5 mm (on three axes, respectively) under a static condition and 28.2, 34.1, and 18.1 mm under a dynamic condition. Such accuracy might be acceptable for a TV, but the position error would have to be minimized for tablets. Seongkook Heo and his colleagues admitted that the IrCube has a fundamental limitation due to reflection.[7] With a tabletop computer, when the pointer approached a surface, reflection from that surface caused the pointer position to disappear.

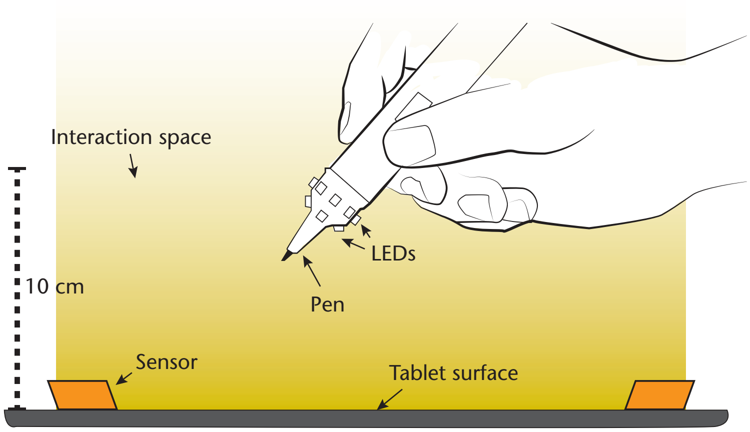


Figure 1. The IrPen interaction space. The IrPen extends the IrCube tracking system by taking into account requirements specific to tablet computers.

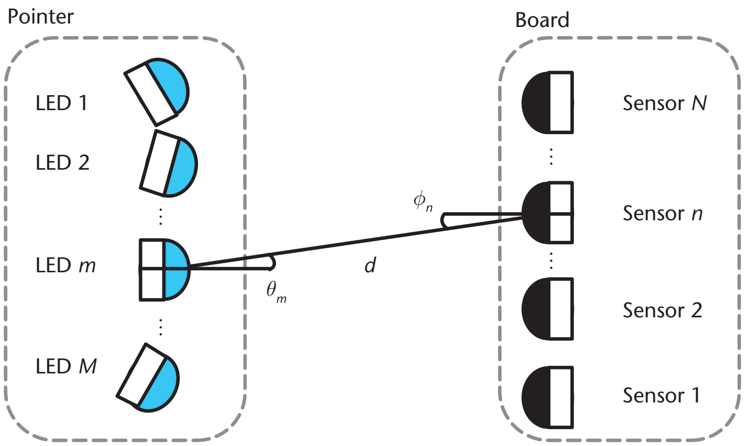


Figure 2. The forward model for the IrCube tracking system.[7] It explains a sensor's output when the distance and angle between an LED and the sensor is given. *d* is the distance between the *m*th LED and the *n*th sensor; θ_n and ϕ_m are the angles between the sensor and LED relative to each other.

The IrPen

The IrPen comprises a pen and a sensor frame on a Samsung Series 7 slate, with an 11.6″ display (see Figure 3a). We implemented the tracker software in C#.

The Pen

The pen contains 12 LEDs, a pen tip, a pressure sensor, and a button (see Figure 3b). First, the LEDs turn on and off together to send a synchronization signal. They then turn on and off sequentially, one at a time. This happens twice, once at full intensity and once at approximately 20 percent of full intensity. This repetition overcomes the sensors's limited dynamic range. The whole process takes approximately 15 ms.

Users press the button to select an object or to enable dragging when the pen is in the interaction space. When the pen touches a surface, its tip pushes the pressure sensor. The pen sends the pressure value and the button's state, encoded in the synchronization signal, to the sensor frame.

The Sensor Frame

The sensor frame contains six sensor modules and a microcontroller. The sensors are at the tablet's four corners and the center of the tablet's top and bottom edges (see Figure 3a). The sensors cover the screen and alleviate occlusion problems. The microcontroller reads sensor output voltages using a 12-bit analog-to-digital converter and sends these values to the computer through a USB connection. A USB bus can power the sensor frame because each sensor module draws less than 30 milliamps.

Signal Processing

The sensor frame's data acquisition time is approximately 15 ms; transmission of a packet from the frame to the tracker software takes approximately 13 ms. A packet consists of 146 integers (12 LEDs × 6 sensors × 2 intensity levels + the button and pressure states). When the system receives a packet, the tracker software requires three signal-processing steps (see Figure 4). Overall, the data frame rate is 35 fps, and these steps' mean computation time is approximately 13 ms with a 3-GHz Intel Core i5-2320 CPU.

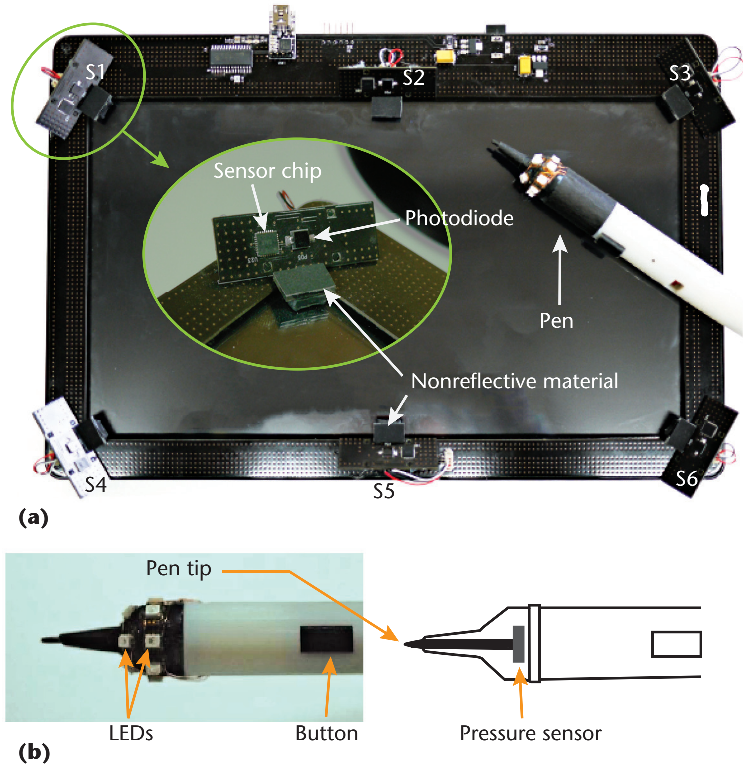


Figure 3. The IrPen. (a) The complete system. (b) The prototype pen. The IrPen employs six sensors (S1–S6) on a Samsung Series 7 slate, with an 11.6″ display.

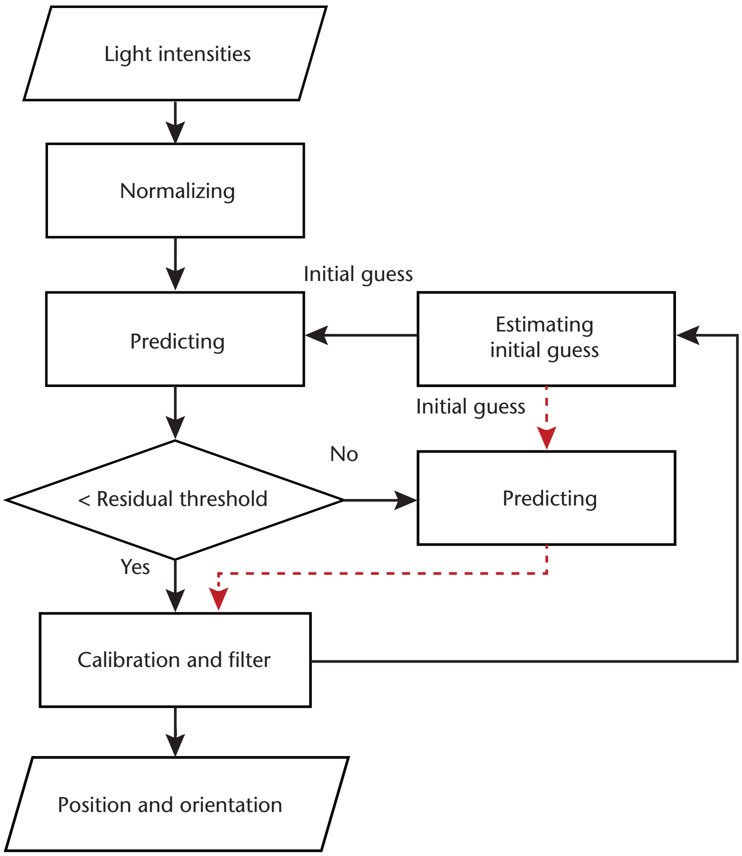


Figure 4. The signal-processing steps. These steps' mean computation time is approximately 13 ms with a 3-GHz Intel Core i5-2320 CPU.

The first step normalizes raw sensor data. This is required because each sensor chip has a different sensing range and sensitivity. We used a linear mapping function for each chip that mapped raw sensor output to a physical light intensity value. We experimentally determined these functions. There are two sensor outputs—one for each LED intensity level. We designed the signal-processing module to select an unsaturated output. If both outputs are unsaturated, the module selects the larger one because it has a higher signal-to-noise ratio. If both outputs are saturated, the module discards them.

In the second step, an optimization algorithm predicts the pen's position and orientation using the normalized values from the first step and an initial guess of the position and orientation. This guess assumes that the pen tip's speed is constant. For optimization, we chose the trust-region reflective algorithm;[8] as it runs, the forward model predicts sensor outputs. If the residual from the optimization exceeds a certain threshold, the algorithm runs again with a different initial guess (as indicated by the dotted lines in Figure 4).

The third step runs an error calibration and a stabilization filter. We included the calibration to increase the IrPen's accuracy because we initially found systematic error in the predictions. We modeled this error using three polynomial functions, one for each axis.

We filter the calibration results to reduce movement from hand tremors. We first tried a low-pass filter to achieve stabilization, but this resulted in latency. To handle a similar problem, a previous study employed an adaptive low-pass filter in which the pointer's average velocity controlled the filter's pole value.[9] The filter effectively compensated for hand tremors and shaking, without causing latency. We applied this filter to our system, on which it was equally effective.

From the IrCube to the IrPen

When moving from the IrCube to the IrPen, we encountered several major technical issues.

Occlusion by the Hand

Because the hand holding the IrPen could block a sensor, the IrPen should be able to tolerate one or two blocked sensors. This is possible because the number of sensor outputs (6 sensors × 12 LEDs = 72 outputs) far exceeds the number of parameters the optimization algorithm requires.

We decided to use six sensors before we considered the occlusion problem. Six sensors were the minimum we thought necessary to cover the

Related Work in 6-DOF Pen Interfaces

Although some Wacom styluses (www.wacom.com) provide six-degree-of-freedom (6-DOF) data, system limitations prevent them from fully supporting interaction techniques that use 6 DOF. For example, their tilt-sensing capability has a limited sensing angle (from –60 degrees to 60 degrees), and they produce much noise. The pens can be tracked as far as 20 mm from a surface, but this isn't likely to support in-air interaction such as Sriram Subramanian and his colleagues described.[1]

To estimate a pen's orientation, David Lee and his colleagues equipped a pen system with a 9-DOF sensor module.[2] The resulting system, however, was susceptible to electromagnetic interference from other devices and couldn't track the pen's position with stability. Similarly, magnetic trackers such as the one in the Polhemus system (www.polhemus.com) are sensitive to electromagnetic interference from other devices. This limitation is critical because electromagnetic interference is highly likely in a mobile environment.

To overcome pen systems' limitations, researchers have employed the Vicon motion-tracking system (www.vicon.com), which uses cameras to track objects. The pattern of Vicon markers on an object serves as a reference points to estimate the target's 6 DOFs. So, the distance between each marker must be large enough for the system to estimate the target's 6 DOFs. Ultrasonic trackers also require a large target to distinguish each transmitter's phase differences. Unlike these systems, our IrPen tracking system supports small targets and thus is applicable to tablet computers (see the main article).

Daniel Taub proposed a system that tracks a pen's 6 DOFs on desktop computers.[3] It uses a small visual tag on the pen and the camera. It employs a pattern-matching algorithm to detect the tag and predicts the pen's 6 DOF by analyzing the tag's size and orientation. It works with a camera-equipped optical touchscreen, but such touchscreens are too thick for tablets.

GaussSense comprises a Hall sensor matrix and a pen with a permanent magnet.[4] This system can precisely detect the pen's position near the surface. However, the magnetic field's strength is inversely proportional to the cube of the distance and decreases rapidly as the angle increases. So, a pen with constant magnetic strength might have a limited tracking range.

Compared to GaussSense, the IrPen provides a wider space for interaction because it uses light instead of magnets. zSpace (http://zspace.com) is a commercial product that tracks the 6 DOFs of the pen and the user's head using four infrared cameras on the display. It provides direct 3D manipulation when the user employs the pen over the display. However, applying this system to tablets would be difficult because the four required cameras might result in high computation costs and power consumption.

References

1. S. Subramanian, D. Aliakseyeu, and A. Lucero, "Multi-layer Interaction for Digital Tables," *Proc. 19th Ann. ACM Symp. User Interface Software and Technology* (UIST 06), 2006, pp. 269–272.
2. D. Lee et al., "PhantomPen: Virtualization of Pen Head for Digital Drawing Free from Pen Occlusion & Visual Parallax," *Proc. 25th Ann. ACM Symp. User Interface Software and Technology* (UIST 12), 2012, pp. 331–340.
3. D.M. Taub, "The BoPen: A Tangible Pointer Tracked in Six Degrees of Freedom," master's thesis, Dept. of Electrical Eng. and Computer Science, MIT, 2006.
4. R. Liang et al., "GaussSense: Attachable Stylus Sensing Using Magnetic Sensor Grid," *Proc. 25th Ann. ACM Symp. User Interface Software and Technology* (UIST 12), 2012, pp. 319–326.

screen. We conducted a simulation to find the best configuration that maximized the number of valid sensor outputs when occlusion occurred. Figure 3a shows the optimal configuration that we determined. In this configuration, the four corner sensors' azimuth angle is approximately 30 degrees, and all sensors have a 45-degree elevation angle.

To evaluate whether this configuration tolerates partial sensor occlusion, we collected sensor data from three scenarios. In the first scenario, we collected data at 1,071 positions (at 1-cm intervals in an $8 \times 16 \times 6 = 768$ cm^3 space) with the pen upright. In the second scenario, we collected data while the pen rotated around the *z*-axis from 0 to 90 degrees. In the third scenario, we collected data while the pen rotated around the *y*-axis from 0 to 90 degrees. We then computed the pen's positions and orientations with no sensors, one sensor, or two sensors occluded. In the latter two cases, we excluded one or two corner sensors because they're most likely to be occluded by the hand.

Table 1. Results for testing the IrPen without and with occlusion by the hand.

	Error distance (cm)			Error angle (degrees)		
Case	***x*-axis**	***y*-axis**	***z*-axis**	***x*-axis**	***y*-axis**	***z*-axis**
Nonoccluded	0.50	0.48	0.72	3.12	1.99	2.19
Sensor 1 is occluded.	0.54	0.52	0.82	2.75	2.62	2.15
Sensors 1 and 3 are occluded.	0.48	0.60	0.83	2.56	2.35	2.17

Table 1 shows the results. The error distances and rotational angles in the occluded cases were similar

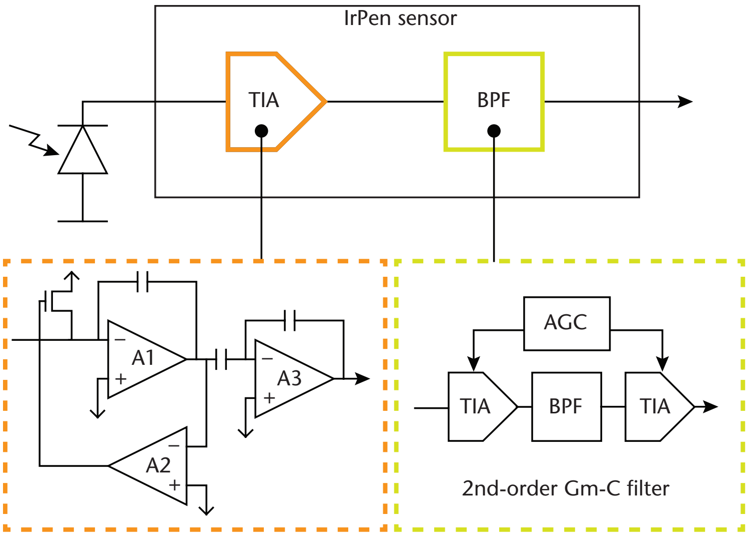


Figure 5. The IrPen sensor's main components are a transimpedance amplifier (TIA) and band-pass filter (BPF). A1, A2, and A3 are op-amps (operational amplifiers); AGC stands for automatic gain control; and GM-C stands for transconductance capacitance.

Table 2. How different forward models affected accuracy.

	Error distance (cm)			Error angle (degrees)		
Forward model	***x*-axis**	***y*-axis**	***z*-axis**	***x*-axis**	***y*-axis**	***z*-axis**
The IrCube	0.62	0.68	0.78	3.82	4.62	4.76
The basic IrPen	0.49	0.47	0.72	3.15	1.99	2.23
The IrPen with pen tip effect	0.50	0.48	0.72	3.12	1.99	2.19
The IrPen with pen tip effect and calibration	0.36	0.25	0.38	—	—	—

to those of the nonoccluded case. This verifies that the IrPen could still accurately determine the pen position with one or two occluded sensors.

Small, Fast Analog Optical Sensors

The IrPen sensors must be able to accurately measure light intensity. They must be fast enough to respond to light modulated at 1 MHz. They also must have a large dynamic range that can cope with varying intensities as the pen moves across a screen. In addition, they must be small enough for a tablet.

Because no off-the-shelf sensors met these requirements, we implemented a dedicated sensor. To meet the IrPen size requirements, we had to implement the sensor as an integrated circuit. The integrated circuit (see Figure 5) has two main blocks: a transimpedance amplifier (TIA) and band-pass filter (BPF).

In the TIA, op-amp (operational amplifier) A1 amplifies the photocurrent through the photodiode. A feedback loop through op-amp A2 avoids saturation of the TIA due to ambient lighting. An additional filter uses op-amp A3 to remove any remaining DC component.

The BPF passes the 1-MHz signal from the pen and removes other noise components. Because we used a CMOS (complementary metal-oxide semiconductor), we chose a second-order transconductance-capacitance filter (Gm-C filter). The best Q factor (quality factor) we could achieve was 40. The BPF had to maintain its gain constant, independent of the input signal's amplitude. This requirement is difficult for a Gm-C filter to satisfy, so we designed the BPF's structure as in Figure 5. The basic idea was to position an amplifier before the BPF and use an automatic gain control to maintain a constant amplitude of the signal to the BPF. Another amplifier after the BPF has an inverse gain; it restores the amplification produced by the first amplifier.

To determine the BPF's success, we examined sensor output under three lighting conditions: in a dark room at night (220 lux), under a fluorescent light (890 lux), and near the window in the daytime (1,100 lux). The six sensors' mean output change was approximately 0.7 percent of the output range. This indicates that the sensors sufficiently rejected ambient light, making the IrPen insensitive to ambient-light changes.

The sensor chip is approximately 6 × 6 mm, which is comparable to the external photodiode's size. The sensor module is approximately 42 × 15 mm, but we can reduce this if necessary, taking into account the chip's and photodiode's size.

Higher Accuracy

Because the IrCube was designed for a TV environment, it assumes all LEDs are at the same position. In that environment, small differences in LED positions are inconsequential because of the interaction space's size.

Because we intend the IrPen for a smaller interaction space, we modified the forward model to include each LED's exact position. This change effectively reduced errors (see Table 2). We also included the effect of the pen tip, which occluded lights from four LEDs from 63 to 90 degrees. This inclusion effectively reduced the orientation error.

We measured the resulting position errors at approximately 5 mm, which is comparable to that of a commercial product (for example, the zSpace system). We calculated these tracking errors with only the raw predicted position and orientation, without calibration or filtering. The tracking errors were more systematic than random and were due mainly to a discrepancy between the system and mathematical model. To compensate for the systematic part of the errors, we used calibration; Table 2's last row shows the results.

Screen Reflection

Screen reflection is problematic because the IrPen interacts close to the screen. Figure 6a illustrates the reflection problem as it applies to the IrCube. As the pointer approaches the screen, the pointer's computed *z*-coordinate initially decreases but increases past a certain point.

To avoid this phenomenon, we placed the LEDs some distance above the pen tip and placed the sensors near the screen. So, the LEDs are usually at least 2 cm from the screen, owing to the pen tip's length. Figure 6b shows the computed height of the pen as it approaches the screen. A nonzero offset error remains but isn't problematic; again, we used a simple calibration to compensate for it. Figure 6c illustrates that the surface area that reflects the LED light is small near the sensor. This area can include the tablet's bezel, which can be made nonreflecting.

Applications

Three use cases illustrate the types of new applications that the IrPen enables.

3D Object Manipulation with Touch Gestures

In this use case, users employed the pen to manipulate objects in a 3D space (see Figure 7a). Because 3D manipulation using a high-DOF tool has been heavily researched, we tried not to replicate the typical scenarios but tried to take it a small step further. Also, although various studies have investigated bimanual interaction that combines pen and touch interfaces, this use case was unique in that it combined a 6-DOF pen and a touchscreen.

By tapping the toggle button on the screen's top left, users switched between the drawing and manipulation modes. In drawing mode, users drew objects directly by dragging the pen in the space. The dragging motion's starting and ending points determined the object's bounding box. In manipulation mode, users selected an object and manipulated it by moving and rotating the pen. In both modes, users could move the workspace by touching the screen with a finger and moving the finger.

Adding Depth to a GUI

This use case involved augmenting a 2D GUI when 6 DOFs are available (see Figure 7b). For example, users could preview a folder's contents when they selected a folder with the pen in the air. The preview's detail decreased as the pen neared the screen, and vice versa. Users opened the folder by rolling the pen. A similar technique worked for documents.

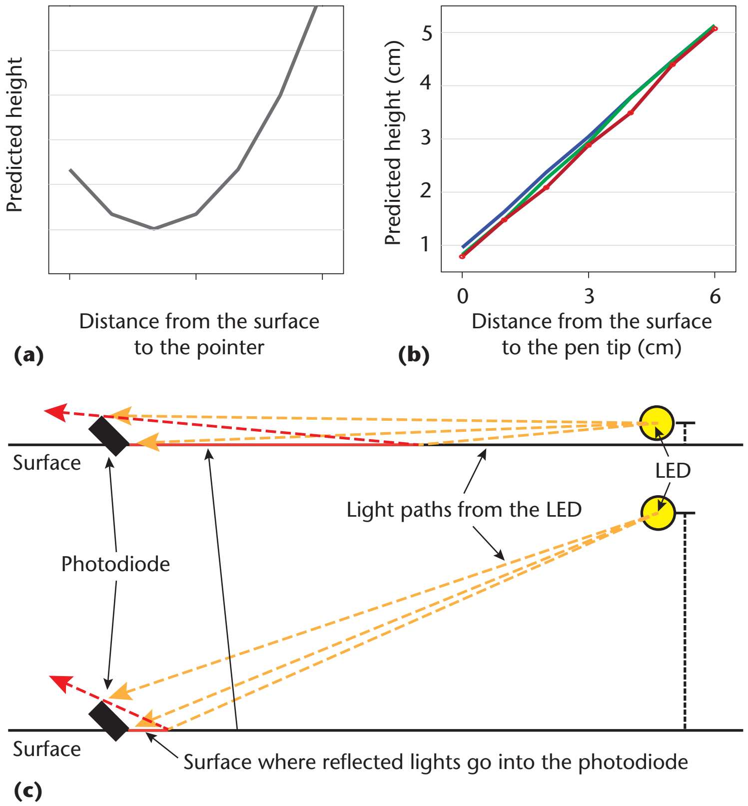


Figure 6. Dealing with the reflection problem. (a) The reflection problem on the IrCube. (b) The computed pen height as the pen approaches the screen, for the IrPen. Each line represents the prediction at a different location: blue indicates the screen's center, red indicates the sensor location, and green indicates the midpoint between the previous two locations. (c) The light paths from the IrPen LEDs to the photodiodes at two distances from the LED to the screen surface.

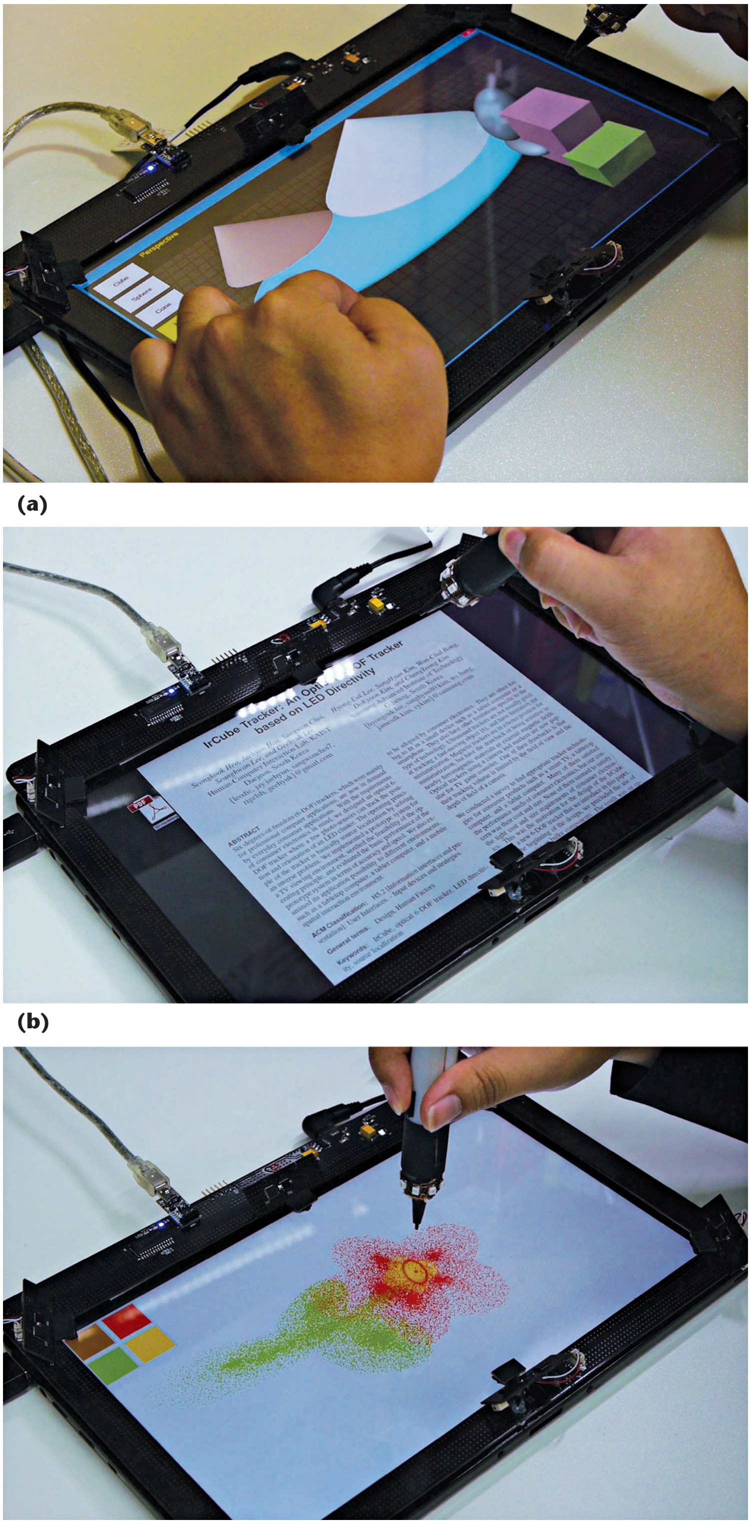
Painting with a Spray Tool

Here, the pen represented a spray tool (see Figure 7c). The tool's footprint changed as the user moved and tilted the pen. The system randomly drew small dots in the footprint. When the pen was upright, the footprint was a circle. When the pen moved up, the footprint became larger, and its shape changed with the pen's tilt angle.

Although the spray tool used only 5 DOFs (it didn't use pen rolling), it demonstrated that multiple DOFs can be useful in simulating hand tools. Most hand tools, such as pencils, carving knives, and flashlights, aren't 2D tools; they're actually 6-DOF tools. Such mimicking of hand tools can give tablet users new ways to interact with their device.

System Evaluations

We had six people use the three applications and then interviewed them. They all noted that the IrPen worked as intended, although they noticed an offset between the pen and the cursor position. None of them reported occlusion problems, even though their hand blocked sensors several times.



(a)

(b)

(c)

Figure 7. Three IrPen applications: (a) 3D object manipulation with touch gestures, (b) adding depth to a GUI, and (c) painting using a spray tool.

They also suggested improvements to the applications' interface, such as a better combination of touch gestures and pen operations and better feedback to indicate the cursor position.

As the participants pointed out, the offset is clearly a limitation of the current prototype. The offset isn't perceptible when the pen is over the screen but becomes perceptible when the pen moves onto the surface. So, we plan to improve the system's precision when the pen is near the surface. This might involve pursuing a more precise forward model and a better calibration model. We're also considering how to combine the IrPen system output with capacitive-touchscreen output so that they complement each other. Another immediate plan is to reduce the pen's power consumption. This is important if the IrPen is to become a practical solution for tablets.

Our research's main goal was to answer the technical problems imposed by a tablet environment, such as handling occlusions, creating small, fast sensor chips, and handling reflection caused by using a touchscreen. The current prototype has a positional accuracy of approximately 0.3 cm and a rotational accuracy of approximately 2.5 degrees. As we mentioned before, the computation time per frame is approximately 13 ms. We found that this accuracy and time are acceptable to users.

Acknowledgments

The IT R&D program of the Ministry of Knowledge Economy / Korea Evaluation Institute of Industrial Technology (grant KI10041244, SmartTV 2.0 Software Platform) supported this research.

References

1. T. Grossman et al., "Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 06), 2006, pp. 861–870.
2. S. Subramanian, D. Aliakseyeu, and A. Lucero, "Multi-layer Interaction for Digital Tables," *Proc. 19th Ann. ACM Symp. User Interface Software and Technology* (UIST 06), 2006, pp. 269–272.
3. X. Bi et al., "An Exploration of Pen Rolling for Pen-Based Interaction," *Proc. 21st Ann. ACM Symp. User Interface Software and Technology* (UIST 08), 2008, pp. 191–200.
4. J.J. LaViola and D.F. Keefe, "3D Spatial Interaction: Applications for Art, Design, and Science," *ACM Siggraph 2011 Courses*, 2011, article 1.
5. B. Baxter et al., "DAB: Interactive Haptic Painting with 3D Virtual Brushes," *Proc. Siggraph*, 2001, pp. 461–468.

6. D. Lee et al., "PhantomPen: Virtualization of Pen Head for Digital Drawing Free from Pen Occlusion & Visual Parallax," *Proc. 25th Ann. ACM Symp. User Interface Software and Technology* (UIST 12), 2012, pp. 331–340.
7. S. Heo et al., "The IrCube Tracker: An Optical 6-DOF Tracker Based on LED Directivity," *Proc. 24th Ann. ACM Symp. User Interface Software and Technology* (UIST 11), 2011, pp. 577–586.
8. J. More and D. Sorensen, "Computing a Trust Region Step," *SIAM J. Scientific and Statistical Computing*, vol. 4, no. 3, 1983, pp. 553–572.
9. G. Lee et al., "A TV Pointing Device Using LED Directivity," *Proc. 2011 IEEE Int'l Conf. Consumer Electronics* (ICCE 11), 2011, pp. 619–620.

Jaehyun Han *is a PhD candidate in computer science at the Korea Advanced Institute of Science and Technology (KAIST). His research interest is human–computer interaction, especially designing interaction techniques on or over surfaces and developing novel input devices. Han received an MS in computer science from KAIST. Contact him at jay.jaehyun@gmail.com.*

Seongkook Heo *is a PhD candidate in computer science at the Korea Advanced Institute of Science and Technology (KAIST). His research interest is designing novel interaction techniques, especially enhancing touch interaction by using a finger's dynamic properties. He received an MS in information and communication engineering from KAIST. Contact him at seongkook@kaist.ac.kr.*

Hyong-Euk Lee *is a senior researcher at the Samsung Advanced Institute of Technology, where he leads a 3D interactive-graphics project. His research interest is human–computer and human–robot interaction, especially realistic graphics rendering and physics simulation for mobile and medical devices, interactive graphics, motion-based interaction, machine learning, and service robots. He's a member of IEEE. Lee received a PhD in electrical engineering from the Korea Advanced Institute of Science and Technology. Contact him at hyongeuk.lee@samsung.com.*

Geehyuk Lee *is an associate professor of computer science at the Korean Advanced Institute of Science and Technology. His primary research interests are interaction devices and techniques for smart information appliances, wearable computers, and smart environments. Lee received a PhD in electrical engineering from the University of Pennsylvania. Contact him at geehyuk@gmail.com.*