Tracking Locations of Moving Hand-held Displays Using Projected Light

Jay Summet & Rahul Sukthankar

summetj@cc.gatech.edu
GVU Center & College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332

rahul.sukthankar@intel.com
Intel Research Pittsburgh
417 South Craig Street, Suite 300
Pittsburgh, PA 15213

Abstract. Researchers have recently demonstrated display positioning using optical sensors in conjunction with temporally-coded patterns of projected light. This paper extends that concept in two important directions. First, we enable such sensors to determine their own location without using radio synchronization signals – allowing cheaper sensors and protecting location privacy. Second, we track the optical sensors over time using adaptive patterns, minimizing the extent of distracting temporal codes to small regions and enabling the remainder of the illuminated region to serve as a useful display while tracking. Our algorithms have been integrated into a prototype system that projects content onto a small, moving surface to create an inexpensive hand-held display for pervasive computing applications.

1 Introduction & Related Work

Augmenting objects in the world with projected computer output is becoming more feasible as projector prices fall and quality improves. Projection screens made of paper, cardboard, or foam core board are so cheap as to be disposable, and could be distributed to visitors at a museum, art gallery or mass-transit system. By carrying one of these display boards under a ceiling-mounted projector, the visitor could access background information about an exhibit, artwork, or train schedule, while the valuable infrastructure (projectors) remains secure from vandalism or theft.

However, projecting output onto objects has traditionally required a time-consuming calibration step, and projecting output onto moving objects has proved to be challenging. Vision systems such as the Visual Panel [9] can track quadrangles suitable for use as projection screens in real time, but difficulty arises when the quadrangle is simultaneously illuminated with dynamic content from a projector. The Hyper-Mask uses active IR-LEDs and an IR-camera to track a white mask and project a character's face on it [8] . The range of that system is limited by the power of the IR-LEDs, sensitivity of the IR-camera and ambient IR illumination.

Recent approaches to localizing objects using active embedded light sensors have greatly decreased the calibration time but not yet achieved projection on moving objects. Raskar *et al.* demonstrated the use of photo-sensitive electronic sensors to locate objects within a projection beam [5]. Single pixel light sensors and radio boards are

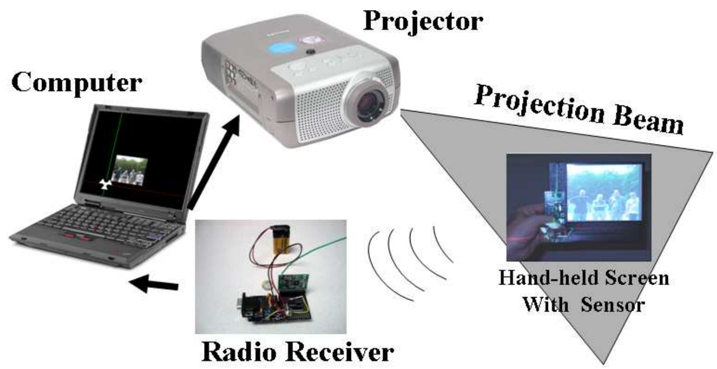

Fig. 1. System Diagram - While moving, a sensor on the hand-held screen detects location information from the projector and broadcasts it. A radio receiver returns this information to the computer, which adjusts the display accordingly to keep the projected image on the screen.

affixed to or embedded within objects of interest. After the projector sends a synchronizing radio signal, the sensors are illuminated by a location-encoding Gray code [2] from the projector, and each sensor determines its location and radios it back to the projector system. Lee *et al.* uses similar technology, replacing the radio with a wired tether, to locate display surfaces within a projector's beam for user output purposes [4]. These previous methods exhibit the following problems:

- **Brittleness to Sensing Errors.** If a light value is received incorrectly due to noise, the calculated location value is incorrect, and no indication of the error is given.
- **Sensor Cost.** Because the Raskar *et al.* wireless sensors require a radio receiver for synchronization, in addition to a transmitter, this increases the cost and power requirements for each sensor. The tethered sensors in Lee *et al.* lack true portability, making them unsuitable for non-laboratory use.
- **Sensor Motion.** The previous approaches assume that the location of sensors does not change, and only needs to be measured once. This precludes using the technique on a mobile hand-held screen.

We aim to address these shortcomings in this work. The remainder of this paper is organized as follows: Section 2 describes our scheme for including error-controlling codes into the projected data pattern, and how this solves the first two problems mentioned above. Section 3 describes our approach to continuous tracking of sensors using projected light, while retaining the majority of the projection surface as a user display (Figure 1). Preliminary quantitative results confirm that our system is capable of reliably tracking relatively slow-moving hand-held display screens and objects.

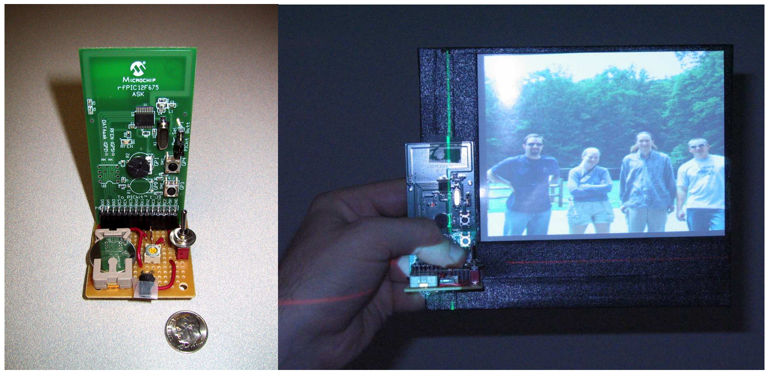

Fig. 2. Left: Optical sensor, attached to rfPIC transmitter board. **Right:** Sensor (under thumb) mounted on the transmitter board (behind thumb), at the bottom left corner of a hand-held projection screen. With one sensor the system tracks the motion of the screen in two dimensions while preserving the display. This allows the image to remain centered on the surface during tracking. With four sensors, the surface can be tracked through arbitrary motions in 3D space.

2 Transmitting Location Data

Our sensor (shown in Figure 2) uses a low cost rfPIC 12F675 micro-controller, with a built in radio transmitter but no receiver, similar to those used in automotive remote keyless entry devices.[1]We used an inexpensive photo diode as a single pixel light sensor. Lee *et al.* showed that optical fibers connected to such sensors could easily be embedded in a white screen and the screen would act as “a light diffuser that helps bounce light into the fiber even at very shallow projection angles” [4].

When a single-pixel optical sensor receives data from a projector (which updates at 60Hz) it records a new intensity value every frame. Our system, like previous work, projects black and white patterns, delivering one bit value (zero or one) per projector frame. In previous systems, the location is encoded using Gray codes. For example, to represent the 1024 unique locations on a 32×32 grid, the independent 5-bit encodings for the X and Y coordinates are concatenated to get $\{0,1,1,0,0,0,1,1,0,0\}$.

Over the period of 10 frames, each of the 1024 different on-screen positions cycles through its own unique code series, producing a unique pattern of light and dark flashes. In this example, a sensor could determine its own location with only 10 projected frames/flashes (1/6th of a second), *if it knew the current offset from the beginning of the code.*

2.1 Error Controlling Code

In previous work, the sensors are either tethered to the projecting computer, making synchronization a non-issue, or a radio signal is used to indicate the beginning of the

[1] Radio receivers are more difficult and expensive to build than transmitters. The rfPIC 12F675 micro-controller we used costs $2.32 USD in quantities over 1600.

projected packet. But, an independent sensor without a radio receiver has no way of determining when a pattern has started.

One way to solve this problem is the inclusion of a framing pattern (which can never appear in a normal location pattern). Unfortunately, because Gray Codes use all possible patterns of ones and zeros, there is no appropriate framing pattern available that is shorter than the localization pattern. Additionally, a framing pattern does not solve the problem of bit errors.

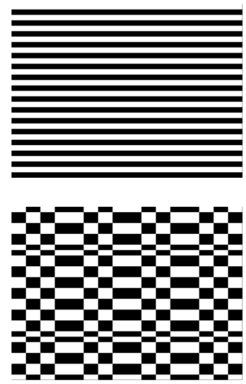

Fig. 3. Single projected frame of Gray Code and check-bit code pattern.

Using a Hamming code[3], SECDED (Single-bit Error Correction Double-bit Error Detection), to transmit the data pattern allows an independent sensor to both synchronize with the data source, as well as detect bit errors. The SECDED code requires the use of $(log_2 N) + 1$ check bits for N data bits. We selected the SECDED code because it was straightforward to implement on an 8-bit micro-controller without floating point math support, and limited processing power and memory. The SECDED code can correct a single bit of error, and detect (but not correct) two corrupted bits. To increase robustness, we used it solely for error detection.

In our implementation, which delivers 16 bits of location information and uses 5 check bits, the SECDED code increases packet size and transmission time by 31%. This reduces our location data speed from a potential 3.75 packets per second to 2.85 packets per second but gives us automatic synchronization and two bits of error detection per 21 bit packet (Figure 4).

2.2 Validating received packets

While receiving bits from the optical sensors, the rfPIC 12F675 micro-controller on our sensor examines the last 21 bits received, attempting to validate the packet. If the SECDED code indicates that a valid packet was received, the sensor knows that it is synchronized with the bit-stream and that no bit errors have occurred. It then decodes the data bits to determine its own (X,Y) location within the projection beam. In our system, the 16 bits are used to deliver 10 bits of location information (a 32×32 grid) and the remaining six bits are used for a projector ID, allowing up to 64 separate projectors to be identified.

Using an XGA projector, our 32×32 grid provides unique locations that are 32×24 pixels in size. The size of the physical area covered by a 32×24 pixel region depends upon the distance of the sensor from the projector, and does not represent a minimum accuracy of our system. If more accuracy is desired, the tracking pattern (in Section 3) can be used to "zero-in" on the sensor, down to a 2×2 pixel level of accuracy for DLP projectors.[2]

[2] Due to automatic spatial dithering in the hardware of DLP projectors, a computer cannot achieve accurate intensity control of pixel groups smaller than 2×2.

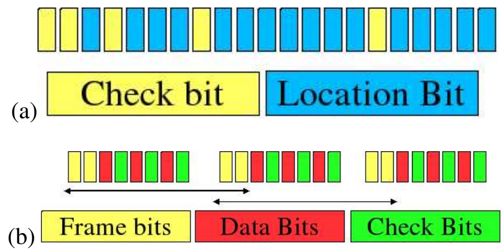

Fig. 4. (a) 21-bit location packet showing 5 check bits and 16 data bits, (b) A stream of three 8 bit tracking packets showing framing bits, data bits, and check bits. Arrows indicate the 10 bit pattern that is decoded, which includes framing bits at each end.

2.3 High Scalability

Because decoding the stream of sensor values is done locally, the only data that needs to be returned to the infrastructure is the successfully-decoded location packet (two bytes, including location and projector ID), and a three-byte sensor ID. In our implementation this is a total of five bytes, which allows 32 projectors, and over 16 million sensors. By adding a few more bytes the number of projectors (and sensors) can be easily expanded.

Local decoding also allows the sensor to activate its radio and return location data only when it has successfully decoded a location packet, saving power and reducing the burden on the shared resource of the RF frequency. Additionally, the sensor knows when the last successful location packet was detected, and its own location, allowing it to take action independent of the infrastructure.

Sensors without on-board decoding must broadcast the data stream continuously, which can pose bandwidth problems over low power RF links, and must rely upon the infrastructure to inform them of their location.

2.4 Independent Operation

In our sample application, the micro-controller transmits its location to the projecting computer, so that the infrastructure can switch to a tracking mode and display content on the hand-held screen attached to the sensor (See Section 3). However, if the sensor were only interested in determining its own location (similar to a GPS receiver), it would not need to divulge its observations to the infrastructure. The Office of the Future project [6] proposes that all lights in an environment will eventually be replaced with projectors, allowing programmable control over the illumination of every centimeter of every surface. The camera included in many phones can be used as a built in optical sensor for our system. If a location-providing infrared projector were mounted over a conference table, a person's mobile phone could switch to silent mode and be able to provide their spouse with location and status information in response to an SMS query, without revealing this information to the infrastructure.

Instead of providing location information directly, the projector could encode other data based upon the location of the optical sensor. For example, a projected electronic classified advertisement board could have a small flashing circle after every telephone number or URL in each advertisement. A user could use a camera phone as a privacy-preserving optical sensor by holding it under the flashing circle of an ad to quickly record the number for a bankruptcy lawyer or mental health support group without revealing its presence to the infrastructure.

3 Tracking

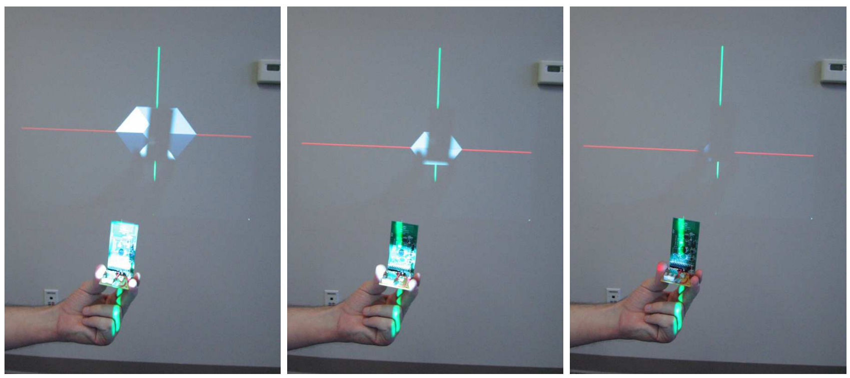

Fig. 5. Three frames showing how our system"zeroes-in" on a near-static sensor. The pattern size was artificially increased at the beginning of this sequence by covering the sensor for two seconds. For purposes of illustration our system projects horizontal (red) and vertical (green) lines which cross at the detected location of the sensor. Note that, in the third frame, the tracking pattern has shrunk to a small spot on the display and that the lines are centered on the sensor.

As with the work by Raskar *et al.* and Lee *et al.*, the projector cannot be used to display application content while it is projecting a full-screen localization pattern. However, once the location of a sensor is detected, our system can switch to a "tracking" mode that projects small localization patterns only around the located sensors and leaves the rest of the projection area available for application display purposes. Additionally, the tracking pattern can be used to "zero-in" on a static sensor, increasing accuracy (Figure 5).

Once the sensor is located, it is only necessary to detect if it moves, and if so, in which direction. Our system does this by projecting a hexagonal pattern with seven distinct areas. The central section covers the sensor if it remains static, and the six surrounding "wedges" indicate the direction of motion that the sensor reports. Identifying these seven areas require only three bits of data to be transmitted in each packet since the projector ID is known from the previously decoded localization packet.

We add two framing bits at the beginning of each packet, as well as three check bits, resulting in an 8-bit packet. We choose to alternate the framing bits of each packet between two zeros {0,0} and two ones {1,1}, enabling us to use both the framing bits from

Speed - Distance (mm)	Recovery Time	Speed - Distance (projector pixels)
73 mm/sec - 314 mm	0.63 sec	74 pixels/sec - 319 pixels
77 mm/sec - 289 mm	0.50 sec	78 pixels/sec - 293 pixels
110 mm/sec - 349 mm	0.53 sec	112 pixels/sec - 354 pixels
128 mm/sec - 320 mm	0.53 sec	130 pixels/sec - 325 pixels

Table 1. Measured successful tracking speeds and recovery times with projector pixels very close to 1×1mm in size. Recovery time is the time from the end of the motion until the tracking system had resolved the sensor's location with the highest level of accuracy available; the sensor's location was known with slightly lesser accuracy throughout the time the sensor was in motion.

the current packet, as well as the framing bits from the following packet to synchronize and detect errors in the transmission channel. The current packet structure allows us to project 7.5 packets per second, which is just enough to track slow hand motions, approximately 12.8 cm/sec when 1.5m from the projector, as discussed below.

In Figure 2 (right) the system is projecting a hexagonal tracking pattern onto the sensor to track its location as it moves. The tracking pattern is intentionally difficult to see, as it is projected on a non-reflective portion of the hand-held screen. The system is using the detected location of the sensor to keep a photograph centered on the reflective display screen attached to the sensor.[3]

Our system employs a quasi-static motion model, which assumes that the sensor performs a random walk from its last reported position. Hence, the new estimate is centered at the previous position and the size of the hexagonal tracking pattern is varied by the confidence of the estimate. The confidence is a heuristic based on the average frequency of location reports and the time since the last reported location, as follows:

– If the system has not received a report for three times the average reporting frequency, it grows the tracker size by a factor of 50%.
– If the system receives a report that is earlier than predicted by the average frequency or late by no more than 10%, it shrinks the tracking pattern by 25% until it reaches a preset minimum size.
– If the system has not received a location report for 2.5 seconds, it assumes that the sensor has been irrecoverably lost, and switches to the global localization pattern.

Figure 5 shows the tracking pattern in the process of shrinking to locate a near-static sensor with greater accuracy.

Table 1 presents four typical "tracked movements," measured with a calibrated video camera, where the sensor is moved from one stable location to another over a period of a few seconds. We chose to test the system with the sensor only 1.5m from the projector. Our absolute speed and distance measures (in millimeters) is specific to our testing setup. At this distance, projector pixels were very close to 1mm in size. The motion distance presented in pixels is a more accurate measure of angular sensitivity of the system, since this is invariant to changes in distance or focal length. For example, if we doubled the size of the projected region by moving the projector away from the

[3] Video: http://www.cc.gatech.edu/~summetj/movies/BurningWell320.avi

sensor, the tracking speed in real units (millimeters) would double (to 25.6 cm/sec at 3m distance from the projector), while the location accuracy would be quartered. However, as the display can be located with no more accuracy than the projector provides, this degradation in accuracy is not a major problem.

4 Alternative Methods

One major advantage of using sensors to detect the optical projector output is that the calibration between the sensor locations (screen) and projector space is obtained directly. Alternative methods for calibrating a projector to a moving display surface involve computer vision using a camera or magnetic motion tracking sensors. The Visual Panel system can track a non-augmented quadrangle screen and translate finger motions over the screen into user interface events but does not demonstrate projecting output on the screen [9]. By augmenting the surface with IR-emitting LED's, the computer vision task is made much easier, but the IR camera and visible light projector must be calibrated [8]. Dynamic Shader Lamps project onto mobile surfaces by using tethered magnetic 6 DOF trackers (affixed to the surface) which are calibrated to the projectors using a manual process [1].

5 Future Work

Figure 2 (right) shows an image projected onto a display surface which is tracked using a single sensor. Using a single sensor allows the surface to translate in two dimensions, but does not detect motion in the Z axis nor rotation. By adding three more photo-diodes to the system (connected to the same micro-controller and radio transmitter) at the other corners of the display surface, an image could be projected upon it as it were moved arbitrarily in 3D.

Additionally, as the board already has an embedded micro-controller and radio transmitter, we intend to further augment the handheld display surface with contact-sensitive film. In addition to returning sensor location reports, the micro-controller could sense and return the location of user touch events on the board's surface, thus developing an extremely inexpensive mobile device that supports user interaction (with the support of environmentally-mounted projectors). Such a board could be manufactured in quantities for $10 to $20 USD, and could be loaned or rented to the public for a negligible deposit.

Currently, the initial locating pattern is very visible and somewhat distracting, covering the entire projection area with a rapidly flashing pattern. This issue could be resolved by encoding the locating pattern in such a way as to be imperceptible to humans. For example, the projector could act as a lamp, throwing an apparently uniform white light which is modulated over time in a manner detectable to a sensor but not a human observer, allowing the system to share the optical sensory channel with humans [7]. Such a coding would slow the initial location acquisition, but could provide a much more user friendly experience.

6 Conclusion

This paper demonstrates a projection system that encodes location data within the projection and a sensor tag which has the following desirable and novel characteristics:

1. Ability to self-synchronize and independently decode location data solely from the optical signal.
2. Robustness to sensing errors due to the use of error detecting codes.
3. Ability to track a sensor while using the remainder of the projection area for graphical output.

Acknowledgments

The Intel Research Pittsburgh lab staff and Ubiquitous Computing Research Group at Georgia Institute of Technology provided valuable feedback on this research. The authors would like to thank Jason Campbell for technical support, Johnny Lee for technical discussions, and Gregory Abowd for editing assistance.

References

1. Deepak Bandyopadhyay, Ramesh Raskar, and Henry Fuchs. Dynamic shader lamps: Painting on real objects. In *The Second IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, 2001.
2. James R. Bitner, Gideon Erlich, and Edward M. Reingold. Efficient generation of the binary reflected gray code and its applications. *Communications of the ACM*, 19(9):517–521, September 1976.
3. Richard W. Hamming. Error detecting and error correcting codes. *Bell Systems Journal*, (29):147–160, April 1950.
4. Johnny C. Lee, Paul H. Dietz, Dan Maynes-Aminzade, Ramesh Raskar, and Scott Hudson. Automatic projector calibration with embedded light sensors. In *Proceedings of UIST 2004*, pages 123–126, 2004.
5. Ramesh Raskar, Paul Beardsley, Jeroen van Baar, Yao Wang, Paul Dietz, Johnny Lee, Darren Leigh, and Thomas Willwacher. Rfig lamps: Interacting with a self-describing world via photosensing wireless tags and projectors. In *Proceedings of SIGGRAPH'04*, volume 23, pages 406–415, 2004.
6. Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. *Computer Graphics*, 32(Annual Conference Series):179–188, 1998.
7. Joshua R. Smith. Imperceptible sensory channels. *IEEE Computer*, 37(6):84–85, June 2004.
8. Tatsuo Yotsukura, Frank Nielsen, Kim Binsted, Shigeo Morishima, and Claudio S. Pinhanez. Hypermask: Talking Head Projected onto Real Object. *The Visual Computer*, 18(2):111–120, 2002.
9. Zhengyou Zhang, Ying Wu, Ying Shan, and Steven Shafer. Visual panel: virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8. ACM Press, 2001.