

# Lumitrack: Low Cost, High Precision, High Speed Tracking with Projected *m*-Sequences

Robert Xiao<sup>1</sup>

Chris Harrison<sup>1</sup>

Karl D.D. Willis<sup>2</sup>

Ivan Poupyrev<sup>2</sup>

Scott E. Hudson<sup>1</sup>

Human-Computer Interaction Institute<sup>1</sup>

Carnegie Mellon University

5000 Forbes Avenue, Pittsburgh PA 15213

{brx, chris.harrison, scott.hudson}@cs.cmu.edu

Disney Research Pittsburgh<sup>2</sup>

4720 Forbes Avenue

Pittsburgh, PA 15213

{karl, ivan.poupyrev}@disneyresearch.com

## ABSTRACT

We present *Lumitrack*, a novel motion tracking technology that uses projected structured patterns and linear optical sensors. Each sensor unit is capable of recovering 2D location within the projection area, while multiple sensors can be combined for up to six degree of freedom (DOF) tracking. Our structured light approach is based on special patterns, called *m*-sequences, in which any consecutive subsequence of *m* bits is unique. Lumitrack can utilize both digital and static projectors, as well as scalable embedded sensing configurations. The resulting system enables high-speed, high precision, and low-cost motion tracking for a wide range of interactive applications. We detail the hardware, operation, and performance characteristics of our approach, as well as a series of example applications that highlight its immediate feasibility and utility.

**ACM Classification:** H.5.2 [Information interfaces and presentation]: User Interfaces - Input devices and strategies.

**General terms:** Design, Human Factors

**Keywords:** Input devices, optical tracking, structured light.

## INTRODUCTION

Movement is one of the fundamental ways we interact with the world around us. We can configure our limbs and bodies into innumerable poses, locomote around the environment, and manipulate objects within it. Unsurprisingly, motion tracking is a fundamental input channel for almost all forms of computing. In response, literally hundreds of approaches have been proposed to capture motion, from accelerometers to Z+RGB depth cameras. The rapidly growing area of embodied and gestural interaction [10] underlines the continuing importance of accurate, fast, and affordable motion tracking technology.

In this paper, we present Lumitrack, a novel sensing technique with several unique qualities. Foremost, our approach is extremely precise: results demonstrate sub-millimeter and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*UIST'13*, October 8–11, 2013, St. Andrews, United Kingdom.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2268-3/13/10...\$15.00.

<http://dx.doi.org/10.1145/2501988.2502022>

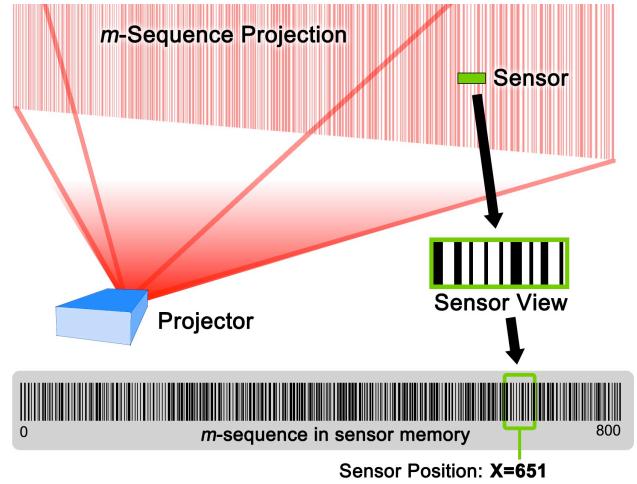


Figure 1. Lumitrack enables high speed, high precision, low-cost tracking by projecting a static binary pattern directly onto a linear optical sensor. A small section of this unique ‘m-sequence’ pattern falls onto the sensor, enabling 1D position to be recovered.

sub-degree translational and rotational accuracy respectively. It also offers extremely high frame rate with low latency: peak speeds over 1000 tracking frames per second (FPS) delivered ~2.5ms behind real time. Moreover, the components of our system are relatively small and low powered (<1 watt), enabling integration into mobile devices. Finally, the necessary components could be very inexpensive: under \$10 parts cost in volume. These qualities make Lumitrack a powerful and intriguing addition to the toolbox of sensors used by HCI researchers and practitioners. The sensing approach can be configured and scaled for use with a range of interactive systems including tangible games, drawing applications, and augmented reality.

Lumitrack has two key components: projectors and sensors (see Figures 1 and 2). Our projectors emit a pattern of static structured light called a binary *m*-sequence [20]. These sequences have a special property: every consecutive subsequence of *m* bits within an *m*-sequence is unique. For example, in the following 7-sequence, the subsequence 1010001 appears exactly once:

00001000110111011000010110110101001111010110011100  
10101111001100011101000101000011001001011100010011

In fact, any consecutive seven digits are guaranteed to only appear once. Our projectors use a 25-sequence where 1's are emitted lines of light and 0's are dark lines (no light).

The second component of our system is a small optical sensor. This must lie within the projection area to function. Because the projected area is big in relation to the sensor, only a small part of the  $m$ -sequence falls onto its surface (Figures 1 and 2). If at least 25 bits of the pattern can be seen (i.e., if the projector is sufficiently close), the sensor can determine where it lies within the projected field by looking up the subsequence's position within the full  $m$ -sequence (a pattern known to both the sensor and projector). This setup allows for 1D position to be resolved.

As we describe in depth later, Lumitrack uses two orthogonal 1D patterns, which allow the sensors to calculate X and Y position. Using more than one sensor, it is possible to resolve X, Y and Z positions, and even roll/pitch/yaw rotation – i.e., full six degree-of-freedom (6 DOF) tracking.

## RELATED WORK

Motion tracking is a fundamental component of almost all interactive systems. Numerous approaches have been developed to sense and track motion using mechanical, inertial, acoustic, magnetic, and radio-based sensing techniques [35]. Most relevant to Lumitrack are optical sensing systems employing natural features, fixed markers, projected markers, structured light, or projections onto sensors.

### Natural Feature Tracking

Natural features in a scene can be tracked using standard cameras to recover 3D information from a series of 2D images, i.e. using Structure from Motion (SfM) [13,27,34]. When combined with Simultaneous Localization and Mapping (SLAM) [9,36], camera pose can be estimated within a dynamically constructed map of the environment. Although powerful, natural feature tracking techniques are computationally expensive and challenging to implement in embedded systems and mobile devices. 已有手段的缺点

### Marker-based Tracking

Marker-based motion tracking systems instrument objects and the environment with physical markers to simplify the tracking process. A camera is used to detect the markers in the scene and estimate the location and, in some systems, the orientation of the marker. Marker-based systems can utilize printed 2D barcodes [7,15], retro-reflective or light-emitting points [5,40], hidden patterns [31,39] or printed patterns [2]. The Bokode system [22] images sections of illuminated lenslet-equipped barcodes using an out-of-focus camera. Finally, low latency, high precision commercial marker-based systems are available, but typically cost thousands of dollars.

### Projected Marker Tracking

Marker patterns can also be dynamically projected onto surfaces in the environment for use with handheld devices [37,38], interactive tabletops [8], and augmented displays [11]. Projected markers can be identified and tracked using standard marker-based tracking techniques. These systems typically hide the obtrusive appearance of the projected

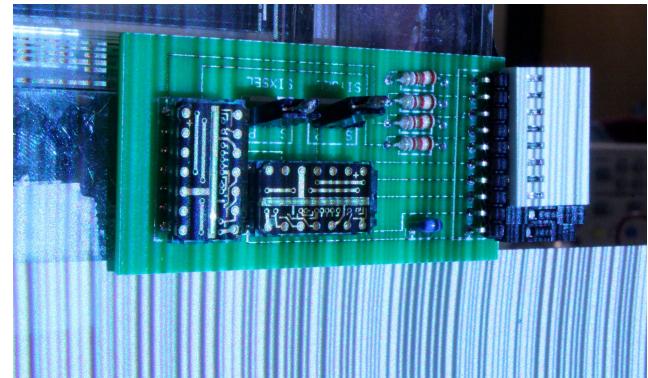


Figure 2. A one-dimensional  $m$ -sequence projected in white onto a Lumitrack sensor. This is analogous to the setup illustrated in Figure 1.

markers using near-infrared projection [19] or temporal multiplexing [11].

### Structured Light Tracking

Projecting geometric patterns on the environment, i.e. *structured light*, allows a camera to infer information about the structure and properties of the environment. A number of structured light schemes have been developed for 3D object digitization and recognition including M-arrays [24], de Bruijn sequences [26], bilinear de Bruijn sequences [16], time-multiplexed grey codes [3], and others [32]. Structured light has been used for interaction to simultaneously capture and display on objects and people [30], to localize mobile devices [17], and in a range of interaction scenarios using structured light depth cameras [12,23]. As structured light systems typically use a camera-projector pair, latency is a common and non-trivial issue [25].

### Projector-Sensor Tracking

Projecting light directly onto an optical sensor enables spatial information to be directly communicated between projector and sensor. The dominant approach has been time-multiplexed projection. For example, in [14], spatial information is temporally communicated to photodiodes embedded in objects using LED clusters. [19] uses a specially-modified projector to project temporal grey code pattern within a monochrome display using an imperceptible change in modulation; sensors within the projection field then read the grey code patterns. PICOntrol [33] modulates the visible output of a projector to transmit time-multiplexed codes onto sensors within the projection field, which activate interactive functions upon detection of an appropriate code.

RFIG Lamps [28] projects time-multiplexed grey codes onto a collection of photodiode sensors. Each sensor computes its position within the projected image using the grey code. A similar approach is taken by Prakash [29], which further extends the sensors to include luminance and orientation sensors. Both systems bear similarities to Lumitrack, but the underlying principles are quite different. Prakash and RFIG Lamps are inherently time-multiplexed, whereas Lumitrack can project unchanging patterns. This opens the

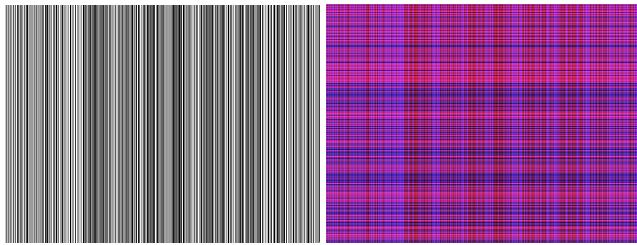


Figure 3. Left: 1D 25-sequence in black and white. Right: Two orthogonal 25-sequences, one rendered in blue and the other in red. Both images have a resolution of 800x600.

possibility of making Lumitrack’s projector configuration much smaller and simpler – Lumitrack requires only a single projector, whereas Prakash requires one projector per bit of spatial resolution.

Although some time-multiplexed systems (e.g., [29]) can perform at comparable speeds to Lumitrack, using a single projected image vastly simplifies projector design. SpeckleSense [41], like Lumitrack, uses static projection for low latency tracking, but is limited to relative translation and distance approximation (whereas Lumitrack offers high frame rate *absolute* tracking). Finally, Bokode [22] can provide absolute tracking by imaging sections of a larger barcode using a lensed camera. In contrast, Lumitrack uses simple, lens-less optical sensors and operates at a significantly higher frame rate.

## PROJECTOR IMPLEMENTATION

The  $m$ -sequence pattern can be projected using a static projector with a fixed pattern or a dynamic digital projector.

### Static Projectors

Static projectors use a light source, focusing optics, and image source to emit a fixed pattern. Due to their simple design, static projectors can be compact, inexpensive, and lightweight. Our prototype static projectors are constructed using LED light sources, off-the-shelf focusing optics, and high-resolution films as image sources. We render our  $m$ -sequences onto black and white film at 8000dpi and insert the film into an off-the-shelf car door “gobo” projector measuring 4.6cm long by 2.2cm in diameter (Figure 4A, 4B). A visible light or invisible infrared LED can be used to power the projection. When projecting in infrared, special consideration needs to be given to the film’s infrared transmission properties. Because many films are designed



Figure 5. The AAXA Technologies L1 projector emits a focus-free image using an LCOS panel and laser light.

to be transmissive to infrared, we use short wavelength 730nm infrared LEDs (Figure 4D) and continue to experiment with different films to maximize image contrast.

For prototyping purposes, we built a visible light, 2D  $m$ -sequence projector using a pair of static projectors (Figure 4C). Although our current prototypes use LED light sources, laser based projectors would have the advantage of focus-free operation provided a suitable  $m$ -sequence diffraction grating or image source could be designed and manufactured.

### Dynamic Projectors

Focus-free dynamic digital projectors are readily available in compact ‘pico-projector’ form factors. Pico-projectors are currently being embedded in devices, such as mobile phones and digital cameras, and offer a unique opportunity for structured light tracking. Coherent laser light, producing crisp images at varying distances, is well suited for human interaction. With laser LCOS projection, a full, single-color image is projected at any instant in time. In contrast, scanning laser projectors rely on the human eye to integrate a single moving point into an image over time. The ‘full-frame’ property of laser LCOS projection allows us to rapidly track the projected  $m$ -sequence.

We use the L1 LCOS laser projector produced by AAXA Technologies [1], which projects time-multiplexed RGBG images at 800x600 resolution. This projection artifact is used to our advantage by displaying the X-axis  $m$ -sequence in the red and blue channels, and the Y-axis  $m$ -sequence in the green channel (Figure 5). By temporally multiplexing orthogonal  $m$ -sequences, we eliminate cross-talk between the projected axes. Using a digital projector allows us to not only track projector motion, but to also render interactive graphics onto the environment around the sensor (Figure

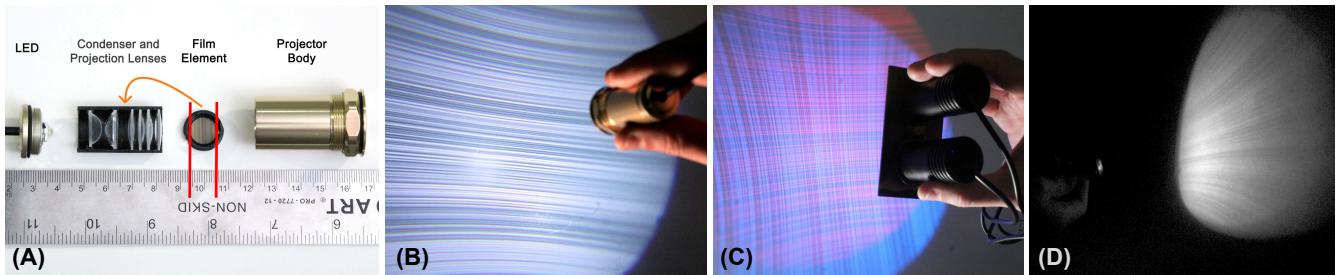


Figure 4. (A): An exploded view of a 1D static projector. (B): Projecting the 1D  $m$ -sequence onto a wall. (C): A two-axis static projector that emits orthogonal  $m$ -sequences, one in red and another in blue. (D): Projecting the 1D  $m$ -sequence in IR.

15). Although we primarily use visible light for prototyping purposes, we ultimately envision the use of two or more frequencies of IR light to hide the tracking patterns.

### Generating $m$ -Sequences

The projected  $m$ -sequences consist of a binary pattern in which every  $m$ -bit subsequence (window) is unique. For tracking robustness, we further constrain our  $m$ -sequences in several ways. We limit the maximum number of consecutive identical bits (a *run* of bits) to three and require that every window contain at least one run of length exactly one. These requirements assist with accurate recovery of the spatial frequency of the pattern received on the linear sensors. Finally, we require that the bit-patterns of different windows differ in at least two places, to ensure that single bit-flips caused by noise could not result in an incorrect identification. To create  $m$ -sequences that fulfill these constraints, we use a sequential generate-and-test approach with backtracking. We use a window size of  $m=25$  for an 800-bit sequence suitable for an 800x600 projection resolution. We used the same approach to find two separate 800-bit sequences with no windows in common, opening the possibility of sequence switching as an additional information channel.

### SENSOR IMPLEMENTATION

The Lumitrack sensor unit consists of a custom printed circuit board connected to a microcontroller. The microcontroller processes sensor data into X/Y positions, which are sent to a computer over USB for interactive applications.

#### Sensor Hardware

The sensor unit consists of a pair of TSL202R 128x1 linear optical sensors manufactured by AMS. The two linear optical sensors are mounted perpendicular to each other (Figure 2). This configuration of two 1D sensors has the distinct advantage of enabling 2D sensing, with a quadratic reduction in total pixels relative to a 2D camera sensor (e.g., 128x128 pixels vs. 2x128 pixels). With fewer pixels to read, we can achieve high frame rates. The optical sensors are highly responsive to a broad spectrum of light, including infrared. An ARM Cortex M3-based 72Mhz Maple Mini board is used to run our custom C software [18].

#### Sensor Sampling

The sensor sampling process is initiated by a timer interrupt, which interrupts the main pattern matching process. It uses both of the Maple Mini's analog to digital converters (ADCs) to sample the linear sensors, one ADC per sensor. By initiating both ADCs simultaneously and waiting for them to both return, the system can sample at a very high rate. We further utilize the high-speed sampling options of the Maple's ADCs to sample at a rate in excess of 1 million samples per second. This reduces the CPU time spent in the sampling process and allows for more time to perform signal processing. By using DMA, the speed could be further improved to a theoretical maximum of 5 million samples per second. Reading both sensors completely takes a total of 101  $\mu$ s. The sensor data is read into two arrays of integers; because this process may interrupt the pattern match-

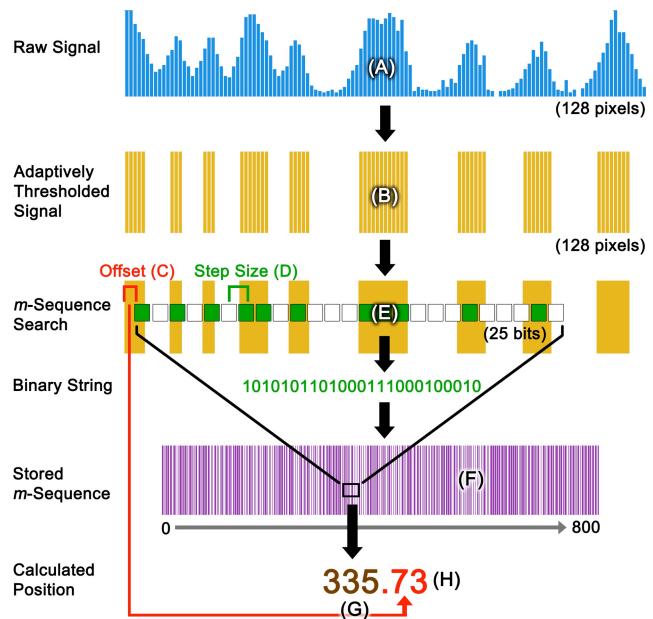


Figure 6. First, our embedded software reads the illumination values from a linear optical sensor (A). This signal is then adaptively thresholded (B). Next, the software exhaustively tests different offsets (C) and step sizes (D), testing to see if the pattern (E) exists in a stored  $m$ -sequence (F). If a sub-sequence match is found, a gross position estimate is yielded (G). The offset (C) provides fractional position accuracy within the  $m$ -sequence (H).

ing process, double-buffering is used to ensure data integrity. The raw data collected from one linear sensor is shown in Figure 6A.

Between each sampling burst, the linear optical sensors accumulate the electrical charges from their photodiode array. This time period, called the *integration time*, is controlled by the timer period. Adjusting the integration time is a key parameter in optimizing the system performance. At close range, the integration time should be short to avoid saturating the sensors, while at long range, the integration time should be longer to improve sensitivity.

#### Adaptive Thresholding and Quick Rejection

For each linear optical sensor, the variance of the analog values is computed. If this value falls below a certain threshold, the system infers that a pattern is not present, and skips the computation. This happens roughly 50% of the time when using our L1 projector due to the projector's color time multiplexing: the pattern aligned with the sensor's orientation is only active half of the time. The quick rejection mechanism allows for more computation time for the other sensor, allowing for a higher overall frame rate.

The sampled analog values (Figure 6A) are adaptively thresholded with a specific window size to produce a binary pixel array (Figure 6B). The thresholding algorithm produces a 1 if the analog value under consideration is larger than the mean across a window centered on that value, and otherwise outputs a 0. The window size is adjustable, and

constitutes another key parameter for performance optimization purposes. At short range, the window size should be small to accommodate a denser arrangement of bits, while at long range, the window size should be large because fewer bits will fall on the sensor.

### m-Sequence Searching

The pattern falling on the linear sensor consists of a subsequence of the  $m$ -sequence. Each bit (projected line) of the sequence covers multiple pixels of the sensor and the step size (number of pixels from one bit to the next) varies based on the distance from the projector to the sensor. For example, in Figure 6, a single unit line falls onto approximately 4 sensor pixels. Furthermore, the offset from the edge of the linear sensor to the start of the first complete bit also varies with the precise position of the projector.

Thus, there are two unknown parameters: the *offset* and the *step size* (Figure 6C and 6D). To determine these two parameters, we run an exhaustive search algorithm over possible step size and offset parameters. We use 16.16 fixed-point arithmetic to enable fine-grained search parameters without the overhead of floating-point computations on the embedded ARM chip. We test step sizes from 2 to 5 pixels in increments of 1/7 of a pixel, and offsets from zero up to the step size in increments of a whole pixel. This results in a total of 63 parameter combinations. For each of these combinations, we sample the thresholded pixel array to obtain an  $m$ -bit integer, which is looked up in a fixed (pre-compiled) hashtable that maps subsequences to their index in the  $m$ -sequence. If the parameter combination is correct, the lookup should succeed (assuming no sensor or sampling errors occur). Note that this lookup is performed for every parameter combination. With our standard 800-bit 25-sequence, there are  $2^{25}$  possible 25-bit sequences and less than  $2^{10}$  valid sequences, so the chances of a false-positive lookup are less than one in  $2^{15}$ . If a lookup succeeds, the Maple Mini sends the looked-up index (i.e., the linear position) and the step size and offset parameters over the USB serial port.

### Sub-unit $m$ -Sequence Tracking Precision

The offset from the sensor's edge to the start of the first complete bit (Figure 6C) offers a way to compute the projector's position with sub-unit accuracy. If the pattern moves by a single pixel on the sensor, the offset will change by one pixel even if the looked-up  $m$ -sequence index remains the same. By dividing the offset by the step size, we obtain a fraction of a position unit, which can be added to the position to obtain a higher-precision position estimate.

### Computer Software

The computer-side software is written in C++ and runs on a Macbook Pro. It communicates via USB with the Maple's emulated serial port at 115200 baud. Because of the serial port's limited bandwidth, a hard limit of 2300 five-byte updates can be received per second; our system very nearly approaches this maximum. The serial port is continually polled by a background thread that pushes received updates into a queue. With multiple connected sensors, each sensor

I will use wifi

gets a background thread; the threads are synchronized to ensure that the data is up-to-date. Interactive applications query the queues every frame to retrieve the most recent data points. In the case of multiple sensors, our code library provides methods for estimating and filtering the 2 DOF, 4 DOF and 6 DOF positions.

### 2 DOF Tracking

With a single sensor, we can resolve an X/Y projected position. These two degrees of freedom can be used in two ways.<sup>1</sup> First, if the projector is held still and not translated, the X/Y positions landing on the sensor can be interpreted as two rotational degrees (pitch and yaw), as seen in the sword demo described later (Figure 13).<sup>2</sup> Alternatively, by translating the device without changing the orientation of the projector or distance to the sensor plane, the X/Y positions are interpreted as translations within the projector's plane and can be translated to X/Y screen-space coordinates, as is done in the spray painting demo described later.

### 4 DOF Tracking

With two sensors, it is possible to track four degrees of freedom: the X/Y center of the tracked points, the angle of the line joining the points relative to the horizontal, and the distance between the points. The inverse of the distance between the points can be used to calculate the approximate distance from the projector to the sensors.

### 6 DOF Tracking

The "holy grail" of tracking is to track all six degrees of freedom – three spatial dimensions and three rotational dimensions. At a bare minimum, three sensors (six linear sensors) are required, but for robustness we chose to use four sensors to provide an over-determined system. Solving for the six degrees of freedom amounts to solving for a 3D (4x4) projection matrix which projects the 3D positions of the four sensors onto the sensed 2D positions within the projector's projected image. This treatment ignores camera distortion parameters (intrinsic parameters), which we found to be minimal with the L1 projector.

To reduce the number of independent variables in the projection matrix, we calibrated and fixed as many variables as possible. The four sensors are arranged in a custom-built, planar mount with tight tolerances (<1mm). This arrangement is suitable for e.g. affixing sensors to the corners of a computer display (Figure 18, right). We calibrated the projector's vertical and horizontal throw precisely to fix the projection matrix's scaling parameters. The projection matrix then becomes  $P = STR$ , where  $S$  is the calibrated standard projection matrix (derived from the projector's throw and aspect ratio),  $T$  is a translation matrix, and  $R$  is a 3D rotation matrix. Because the linear sensors on our custom PCB are slightly offset, we end up with a set of four pairs of equations, one per sensor:  $(Pp_x)_x = q_x$ ,  $(Pp_y)_y = q_y$  where  $p_x$  and  $p_y$  are the 3D positions of the starting end of the  $x$  and  $y$  linear sensors,  $(q_x, q_y)$  is the 2D tracked position reported by the sensor, and  $(P)_x$ ,  $(P)_y$  denote the  $x/y$  coordinates of  $P$ .

With these eight equations and tracked positions, we perform Gauss-Newton iteration to find a solution which min-

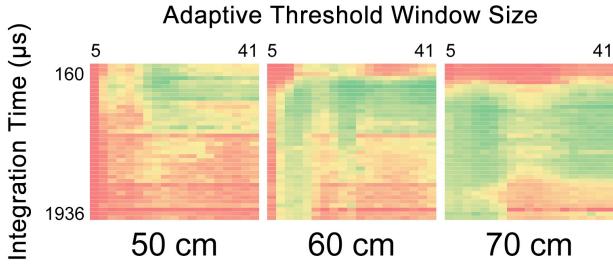


Figure 7. Results from our parameter optimization tests. A red to green scale is used, which indicates low and high tracking rates respectively. Note that as the distance increases, a longer integration time is needed to maintain optimal performance. **因为要搜索更多pattern**

minimizes the total squared error  $\sum \| (Pp_x)_x - q_x \|^2 + \| (Pp_y)_y - q_y \|^2$ , also known as the total squared reprojection error. This yields estimates for the six degrees of freedom. Finally, these estimates are smoothed with a 1€ filter [6] to reduce jitter in the estimates during rapid movement.

#### Parameter Optimization

We performed an experiment to determine the optimal values for integration time and adaptive threshold window size. We wrote a simple Maple mini program to step through the possible combinations of parameters. For each combination, the program performed pattern detection for 500ms and reported the number of successful X data points captured within that time. We ran the program at three distances to the sensor, 50cm, 60cm, and 70cm. The results are summarized in Figure 7. At longer distances, higher integration times are generally more advantageous. Using the data collected by this approach, future implementations could automatically select appropriate sensing parameters to achieve optimal performance over a wider range of distances and lighting conditions. For fixed window sizes, we found that a window size of 23 works well across distances.

#### SYSTEM PERFORMANCE AND EVALUATION

To quantify and convey the strengths of our system, we conducted a series of experiments that sought to evaluate key aspects of our system, including tracking precision, frame rate, latency, and power consumption.

#### Tracking Accuracy

To evaluate the spatial accuracy of Lumitrack, we attached our sensor onto a commercial grade, two-axis CNC platform, having a step size of 1 mil (0.0254mm). We mounted

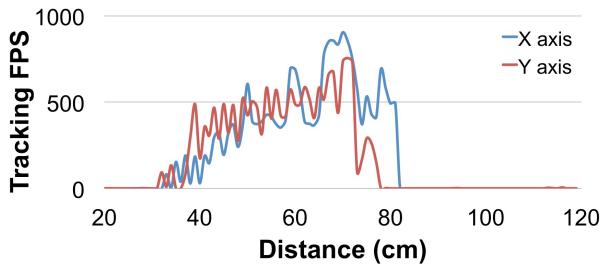


Figure 9. System tracking response for each axis at varying distances between sensor and projector.



Figure 8. Points recorded by sensors during raster pattern scan.

the L1 projector 65cm above the sensor plane and configured it to emit our standard  $m$ -sequence pattern.

We created a custom control program to repeatedly move the platform to a random location within the projected field. The program recorded values streaming from the sensor for one second, which were saved for later analysis. We ran this test for 14 hours, collecting data for 8400 random locations (trials). In total, we collected more than 5.6 million tracking frames. We discarded 221 trials that had no tracking frames for one or both axes, and 139 trials that had a measurement error greater than 25mm (errors of this magnitude were most likely caused by error frames).

Across the remaining 8040 trials, the average Euclidean distance error was 0.532mm (SD = 0.616mm). Except for 14 outliers, the maximum error was 1.297mm. At the testing distance of 65cm, the average error translates into a rotational accuracy of 0.0469°. For comparison, [29] achieved a mean error of 5.8mm (at 300cm) at ~500 FPS.

We also instructed the CNC platform to trace a 1.5cm raster pattern, also at a distance of 65cm. This took 136.0 seconds, during which time our sensor continuously collected 109,363 position values (an average of 804 tracking FPS). The raw data is plotted in Figure 8.

#### Effects of Distance on Tracking Frame Rate

To evaluate the tracking frame rate performance of Lumitrack at various projector-sensor distances, we used the CNC platform to move the projector away from the sensor by increments of 1.0cm. At each stop, the control program recorded the values streaming from the sensor for one se-

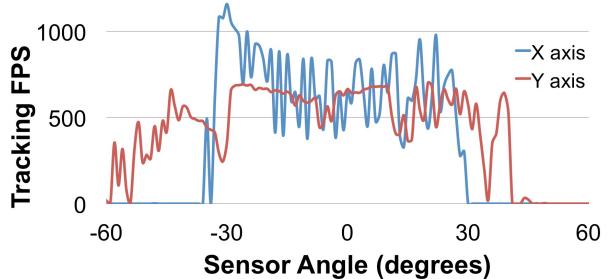


Figure 10. System tracking response for each axis at varying angles between sensor and projector.

cond. Figure 9 shows the number of X-axis (blue) and Y-axis (red) tracking frames obtained in one second as a function of distance. Note that the frame rates in Figures 9 and 10 are  $\frac{1}{2}$  maximum due to temporal multiplexing of the projector.

Note that the sensor parameters used in this experiment were set to the optimal parameters for a distance of 70cm; nevertheless, Lumitrack was capable of tracking at over 500 total tracking frames per second at distances of 40 to 75cm. Adaptive integration time and windowing will help to increase this range further in future work.

### Effects of Rotation on Tracking Frame Rate

We also evaluated the tracking frame rate of Lumitrack under three rotational conditions: sensor rotation, projector pitch, and projector yaw. In each condition, we affixed the rotating component to a rotary table with a step of  $1/80^\circ$ .

In the first condition, the sensor was rotated on the platform while the projector remained fixed. The sensor was oriented so that the Y-axis linear sensor was perpendicular to the table, and the X-axis linear sensor was parallel to the rotating surface. The results, summarized in Figure 10, show that the perpendicular Y-axis sensor picked up frames over a wide range of angles (from  $-60^\circ$  to  $40^\circ$ ), whereas the X-axis sensor's range was slightly more limited ( $-35^\circ$  to  $30^\circ$ ).

In the second and third conditions, we mounted the projector to the rotary table and kept the sensor fixed. We observed that the sensor picked up tracking frames at near maximum speed as soon as the edge of the projection covered the sensor, and so we conclude that the field of view in this case is constrained only by the projector.

### Latency

The total system performance is affected by latency at multiple levels. With the time-multiplexed image from the AAXA projector, only one coordinate (X or Y) is tracked at any given instant of time. Thus, the average latency to obtain sensor reads on both coordinates is equal to half the time spent displaying a single color frame, or around 2 ms (0.5/240 s) for the AAXA projector.

The tracker's sensor latency due to the integration time is around 440 $\mu$ s on average. Processing the sample takes an average of 1250 $\mu$ s on the Maple, so the average processing latency is 1900 $\mu$ s (half of the time to process the previous sample, plus the time to process the current sample). Finally, the data are sent over a serial connection running at 115200 baud. Since this connection can receive at most

# Sensors (DOF)	User w/ Sensor	User w/ Projector
<b>One</b> (2D position)	Spray painting (Figure 11)	Draw App (Fig. 12) Sword (Figure 13)
<b>Two</b> (3D pos. + Z rotation)	Airplane Yoke (Fig. 14) AR Spaceship (Fig. 15)	Highway escape (Figure 16)
<b>Four</b> (3D Pos. + X/Y/Z rot.)	Avatar head (Figure 17)	Helicopter (Figure 18)

Table 1. Categorization of example applications based on sensor-projector configuration and number of sensors.

2000 positions per second, it contributes up to 500 $\mu$ s of latency, or an average of 250 $\mu$ s. Thus, the input latency immediately due to our system is around 2600 $\mu$ s (2.6ms), which is more than enough to provide a fluid interaction

The latency of the sensor configuration is negligible compared to the latency of the operating system and application, and is likely to be imperceptible to an end user [25]. By comparison, other sensing technologies such as the Microsoft Kinect [21] operate at much lower frame rates (e.g., 30 FPS), resulting in noticeable input delays.

### Error Frames

During the tracking accuracy analysis, around 7500 tracking frames belonging to the high-error trials were discarded. Not all frames in these trials were erroneous frames, but we can infer from this that the error rate is at most 1 in 750 frames. During the raster pattern test, 50 erroneous positions were identified by a post-hoc analysis of the data. This is an error rate of 1 in 2000 frames. Thus, the true error rate lies between 0.05% and 0.13% of all frames.

### Power Consumption

The power consumption of the sensor, including microprocessor, is about 55mA at 5V, or around 275mW. The microprocessor's power consumption is about 35mA alone, and no special effort has been made to decrease this. We believe that this could be substantially improved with microprocessor power optimizations and low-power sensor components. The dual static projector configuration shown in Figure 4C consumes about 600 mW. In order to function with high ambient brightness or at a larger distance, more power would be required. The AAXA L1 projector we used consumes 7.5W.

## EXAMPLE APPLICATIONS

To showcase our system's capability and utility, we developed a series of example applications. We categorize the application space based on two criteria: *Projector/sensor configuration* and *number of sensors*. Projector/sensor configuration denotes whether the sensor is held/worn by the user and the projector is mounted in the environment or vice versa [41]. As both the sensors and projector can be made lightweight and portable, either approach is viable for real-time tracking. The number of sensors used determines the degrees of freedom possible for tracking. Table 1 provides a taxonomic overview of our demo applications. Please also see the Video Figure for demonstrations.

### One Sensor (2 DOF)

#### User with Sensor

We modified an off-the-shelf spray paint can, replacing the paint nozzle with a Lumitrack sensor (Figure 11). A button was provided to spray digital paint onto a computer monitor by toggling the transmission of tracking frames. The projector was mounted to the top of the monitor, providing a 2D tracking volume in front of the display.

#### User with Projector

In the first example application, the user manipulated the projector much like a laser pointer, controlling a cursor on the laptop screen (Figure 12) by translating or rotating. But-



Figure 11. Digital spray-painting example application.

tons on the projector allowed the user to toggle between three different  $m$ -sequence patterns which had no  $m$ -bit subsequences in common. The sensor was aware of all three patterns, so in addition to being able to compute its X/Y position within the projected pattern, it was also able to know what  $m$ -sequence was active. We bound these three patterns to a pen, eraser and cursor tool, enabling a basic, free-space painting application.

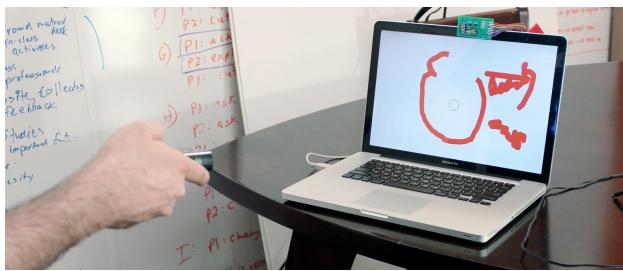


Figure 12. Example painting application. Users can translate and rotate the projector to draw and erase; modes are switched using buttons on the projector.

We used a sword prop for our second example application, which was augmented with a projector (Figure 13, left). On-screen, a virtual sword tracked with the user's rotational movements (Figure 13, Right). One could imagine using this in a sword dueling game requiring high precision and fast reaction times.



Figure 13. Tracking a projector-augmented sword.

## Two Sensors (4 DOF)

### User with Sensor

We fabricated an airplane yoke with sensors embedded in the upper grips (Figure 14). Like its real-world counterparts, this yoke could control the pitch (nose up/down via pull/push inputs) and roll (left/right bank via left/right turn inputs) of a virtual plane.



Figure 14. Lumitrack-instrumented airplane yoke used to control a virtual plane.

We also created a projected augmented reality application. A small spaceship model with fitted with two sensors (Figure 15). We affixed our L1 projector above a play space, facing downwards, blanketing an area with the two orthogonal  $m$ -sequences needed for tracking (Figure 15, right). Additionally, we took advantage of its dynamic projection capabilities to superimpose afterburner graphics and plasma weapon fire. The user can also steer the ship; a star field with parallax is rendered in the background to provide the illusion of directional travel.

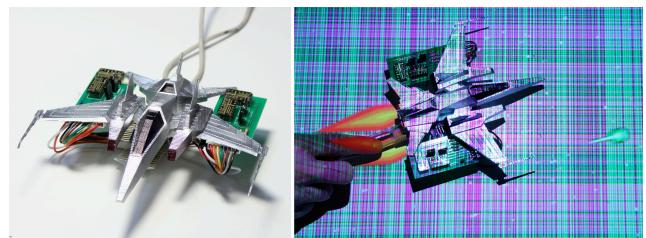


Figure 15. An example augmented-reality application. In addition to projecting  $m$ -sequences, the spaceship model (left) is augmented with afterburner and weapon graphics (right).

### User with Projector

We 3D printed a small car model to sit top of our L1 projector. This car could be translated on a table's surface in front of an augmented laptop running a simple racing game (Figure 16). The player's onscreen car mirrored the user's real-world movements. To capture this movement, two sensors were placed on the front bezel of the laptop.

## Four Sensors (6 DOF)

### User with Sensor

We created a four-sensor rig that could be worn on the user's head. This provided the 3D position and pose of the face. As a proof of concept, we created an avatar head (Fig-



Figure 16. We created a highway racing game where users can translate a car prop on the tabletop to control their in-game car.

ure 17) that followed the user’s position by rotating and translating. The avatar also mirrored the Z-rotation of the user’s head (i.e., side-to-side tilting of the head).

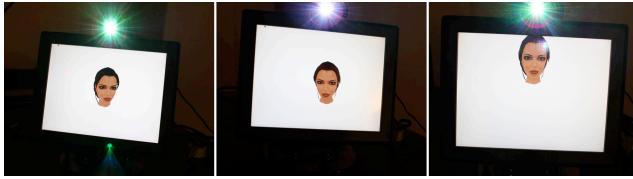


Figure 17. Using 6 DOF tracking of the user’s head, we created an avatar head that always faces the user.

#### User with Projector

As a final demo, we affixed a projector to the sleds of a 3D printed helicopter. This model could be moved and rotated in three dimensions, which was reflected by an onscreen virtual helicopter. In a flight simulator experience, this could be used control a virtual helicopter (e.g., tilting forward for forward movement).

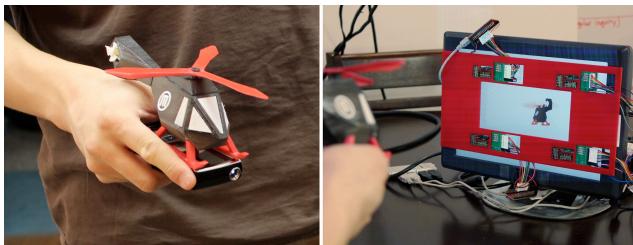


Figure 18. A projector-augmented helicopter model, which could be used to control an on-screen helicopter with six-degrees of freedom.

#### LIMITATIONS AND DRAWBACKS

*Ambient Light.* As with all optical systems, Lumitrack is sensitive to changes in ambient light in the environment. Performance degrades sharply when a significant amount of ambient light is present. We envision increases in projector brightness and optical sensor response will improve the robustness of our system in varying lighting conditions.

*Working Volume.* The working volume in which interaction can take place is limited by a number of factors including the projector brightness, throw ratio, projection angle, and pattern density. Increasing the tracking distance requires a combination of brighter projection, larger sensors, and higher projection resolution, while reducing the minimum tracking distance requires higher sensor resolution. Our evaluation results demonstrate Lumitrack can be used for motion tracking within the spatial range of many typical applications.

*False Positives.* The use of parameter search to locate the  $m$ -subsequence renders the system prone to false positives – resulting in unwanted skips and jumps in the tracked position. These are made relatively rare by selecting a large  $m$  value (window size), but can still occur. Denser patterns with correspondingly higher  $m$  values, combined with more intelligent  $m$ -sequence search approaches, can help mitigate this problem.

*Multi-User Scenarios.* In our current approach, the sensor unit is unable to sense position if multiple projectors overlap, restricting use to single-user applications. Time-multiplexing, based on sensed proximity or other means, could resolve this issue and is the focus of future work.

*Laser Safety.* Depending on the sensing configuration, laser light can potentially be exposed to the user’s eyes. LCOS-based laser projectors, however, are significantly safer than their scanning laser counterparts [4]. The use of IR laser light may require modifications to increase the projector throw ratio and minimize light exposure.

#### CONCLUSIONS AND FUTURE WORK

We introduced Lumitrack, a novel technology for high speed, high precision, low-cost tracking. In future work we envision extending the Lumitrack system in several ways. Firstly, developing color + infrared projectors will enhance the user experience and enable greater interaction with projected content. Secondly, optimizing our  $m$ -sequences to consider higher density encoding schemes (e.g., transition based schemes) and multi-resolution arrays that could enable increased range. Additionally, higher resolution pico-projectors are slated for release (e.g., HD resolution), which will offer improved tracking. Simultaneously, higher resolution linear optical sensors are already on the market (e.g., the TSL1402R is 400 PPI). Finally, we aim to reduce the size of our sensor unit by half to enable integration into a wider range of devices and objects.

#### ACKNOWLEDGMENTS

Funding for this work was provided in part by a Google Ph.D. fellowship, a Qualcomm Innovation Fellowship, NSERC, and NSF grant IIS-1217929.

#### REFERENCES

1. AAXA Technologies. <http://www.aaxatech.com/>
2. Anoto Digital Pen. <http://www.anoto.com/>
3. Bergmann, D. New approach for automatic surface reconstruction with coded light. In *Proc. Remote Sensing and Reconstruction for Three-Dimensional Objects and Scenes*. Vol. 2572, SPIE, 1995, 2-9.
4. Buckley, E. Eye-safety analysis of current laser-based LCOS projection systems. *J. Soc. Information Display*. 18(12), 2010, 1051-1057.
5. Cao, X., Forlines, C. and Balakrishnan, R. Multi-user Interaction Using Handheld Projectors. In *Proc. UIST ’07*, 43-52.
6. Casiez, G., Roussel, N., Vogel, D. 1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *Proc. CHI ’12*, 2527-2530.
7. Celozzi, C., Paravati, G., Sanna, A. and Lamberti, F. A 6-DOF ARTag-based tracking system. *IEEE Trans. Consumer Electronics*, 56(1), February 2010, 203-210.
8. Chan, L.-W., Wu, H.-T., Kao, H.-S., Ko, J.-C., Lin, H.-R., Chen, M.Y., Hsu, J. and Hung, Y.-P., Enabling Beyond surface Interactions for Interactive Surface with an Invisible Projection. In *Proc. UIST ’10*, 263-272.

9. Davison, A., Reid, I., Molton, N. and Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 29(6), 2007, 1052-1067.
10. Dourish, P. Where the Action is: The Foundations of Embodied Interaction. MIT Press, Cambridge, Mass., 2001.
11. Grundhöfer, A., Seeger, M., Hantsch, F. and Bimber, O. Dynamic Adaptation of Projected Imperceptible Codes. In *Proc. ISMAR '07*, 1-10.
12. Harrison, C., Benko, H. and Wilson, A. D. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proc. UIST '11*, 441-450.
13. Hartley, R. I. and Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press. 2004.
14. Heo, S., Han, J., Choi, S., Lee, S., Lee, G., Lee, H., Kim, S., Bang, W., Kim, D. and Kim, C. IrCube tracker: an optical 6-DOF tracker based on LED directivity. In *Proc. UIST '11*, 577-586.
15. Kato, H. and Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. IWAR '99*, 85-94.
16. Kawasaki H., Furukawa R., Sagawa R. and Yagi Y. Dynamic scene shape reconstruction using a single structured light pattern. In *Proc. CVPR '08*, 1-8.
17. Köhler, M., Patel, S.N., Summet, J.W., Stuntebeck, E.P. and Abowd, G.D. TrackSense: infrastructure free precise indoor positioning using projected patterns. In *Proc. Pervasive '07*, 334-350.
18. Leaf Labs. <http://leaflabs.com>
19. Lee, J., Hudson, S. and Dietz, P., Hybrid Infrared and Visible Light Projection for Location Tracking. In *Proc. UIST '07*, 57-60.
20. MacWilliams, F. J. and Sloane, N. J. A. Pseudorandom sequences and arrays. *Proceedings of the IEEE*. 64(12), 1976, 1715-1729.
21. Microsoft Kinect. <http://www.xbox.com/kinect>
22. Mohan, A., Woo, G., Hiura, S., Smithwick, Q., and Raskar, R. Bokode: imperceptible visual tags for camera based interaction from a distance. In *Proc. SIGGRAPH '09*, Article 98, 8 pages.
23. Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A. and Gellersen, H. Interactive Environment-Aware Handheld Projectors for Pervasive Computing Spaces. In *Proc. Pervasive Computing '12*, 197-215.
24. Morita, H., Yajima, K. and Sakata, S. Reconstruction of Surfaces of 3-D Objects by M-array Pattern Projection Method. In *Proc. ICCV '88*, 468-473.
25. Ng, A., Lepinski, J., Wigdor, D., Sanders, S. and Dietz, P. Designing for low-latency direct-touch input. In *Proc. UIST '12*, 453-464.
26. Pagès, J., Salvi, J. and Forest J. A new optimised De Bruijn coding strategy for structured light patterns. In *Proc. CVPR '04*, 284-287.
27. Pollefeys, M., Gool, L. V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J. and Koch, R. Visual modeling with a hand-held camera. *International Journal of Computer Vision*. 59(3), 2004, 207-232.
28. Raskar, R., Beardsley, P., van Baar, J., Wang, Y., Dietz, P., Lee, J., Leigh, D. and Willwacher, T. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. In *Proc. SIGGRAPH '04*, 406-415.
29. Raskar, R., Nii, H., de Decker, B., Hashimoto, Y., Summet, J., Moore, D., Zhao, Y., Westhues, J., Dietz, P., Inami, M., Nayar, S., Barnwell, J., Noland, M., Bekaert, P., Branzoi, V. and Bruns, E. Prakash: lighting aware motion capture using photo-sensing markers and multiplexed illuminators. In *Proc. SIGGRAPH '07*, Article 36, 11 pages.
30. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L. and Fuchs, H. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proc. SIGGRAPH '98*, 179-188.
31. Saito, S., Hiyama, A., Tanikawa, T. and Hirose, M. Indoor Marker-based Localization Using Coded Seamless Pattern for Interior Decoration. In *Proc. VR '07*, 67-74.
32. Salvi, J., Fernandez, S., Pribanic, T. and Llado, X. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, vol. 43, 2010, 2666-2680.
33. Schmidt, D., Molyneaux, D. and Cao, X. PICOntrol: using a handheld projector for direct control of physical devices through visible light. In *Proc. UIST '12*, 379-388.
34. Shiratori, T., Park, H., Sigal, L., Sheikh, Y. and Hodgins, J. Motion capture from body-mounted cameras. In *Proc. SIGGRAPH '11*, Article 31, 10 pages.
35. Welch, G. and Foxlin, E. Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *IEEE Comput. Graph. Appl.* 22(6), November 2002, 24-38.
36. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K. and Colucci, D. The HiBall tracker: High performance wide-area tracking for virtual and augmented environments. In *Proc. VRST '99*, 1-10.
37. Wienss, C., Nikitin, I., Goebels, G., Troche, K., Göbel, M., Nikitina, L. and Müller, S. Sceptre: an infrared laser tracking system for virtual environments. In *Proc. VRST '06*, 45-50.
38. Willis, K.D.D., Poupyrev, I., Hudson, S. E. and Mahler, M. SideBySide: Ad-hoc Multi-user Interaction with Handheld Projectors. In *Proc. UIST '11*, 431-440.
39. Willis, K.D.D., Shiratori, T. and Mahler, M. HideOut: Mobile Projector Interaction with Tangible Objects and Surfaces. In *TEI '13*.
40. Woltring, H. New possibilities for human motion studies by real-time light spot position measurement. *Biotelemetry*, 1(3), 1974, 132-146.
41. Zizka, J., Olwal, A. and Raskar, R. SpeckleSense: fast, precise, low-cost and compact motion sensing using laser speckle. In *Proc. UIST '11*, 489-498.