HoloFlex: A Flexible Light-Field Smartphone with a Microlens Array and a P-OLED Touchscreen

Daniel Gotsch **Xujing Zhang** **Juan Pablo Carrascal** **Roel Vertegaal**
Human Media Lab, Queen's University
Kingston, ON, Canada, K7L 3N6
{goc, xzhang, jp, roel}@cs.queensu.ca

ABSTRACT
We present HoloFlex, a 3D flexible smartphone featuring a light-field display consisting of a high-resolution P-OLED display and an array of 16,640 microlenses. HoloFlex allows mobile users to interact with 3D images featuring natural visual cues such as motion parallax and stereoscopy without glasses or head tracking. Its flexibility allows the use of bend input for interacting with 3D objects along the *z* axis. Images are rendered into 12-pixel wide circular blocks—pinhole views of the 3D scene—which enable ~80 unique viewports at an effective resolution of 160 × 104. The microlens array distributes each pixel from the display in a direction that preserves the angular information of light rays in the 3D scene. We present a preliminary study evaluating the effect of bend input vs. a vertical touch screen slider on 3D docking performance. Results indicate that bend input significantly improves movement time in this task. We also present 3D applications including a 3D editor, a 3D *Angry Birds* game and a 3D teleconferencing system that utilize bend input.

Author Keywords
Organic User Interfaces; Light-field Displays; 3D Input.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
In the real world, humans rely heavily on a number of 3D depth cues to locate and manipulate objects and to navigate their surroundings. Among these depth cues are stereoscopy, provided by the different lines of sight offered by each of our eyes, and motion parallax: the shift of perspective when a viewer changes their relative position. While 3D graphic engines have come a long way in the last half-century, to date much of the 3D content remains rendered as a 2D image on flat panel displays that are unable to provide proper depth cues. One solution for rendering images in 3D is to use a lenticular display, which offers limited forms of glasses-free stereoscopy and limited one-dimensional motion parallax [40]. Other, more immersive solutions, such as the Oculus Rift [41] and the Microsoft HoloLens [38], offer full stereoscopy and motion parallax but require headsets or motion tracking. Recently, there has been a renewed interest in 3D displays that do not require glasses, head tracking or headsets. Such *light-field* displays render a 3D scene while preserving all angular information of the light rays in that scene. With some notable exceptions [28], the size of light-field displays makes them unsuitable for mobile use. Current smartphones are also limited in terms of 3D input. Z-Input—control along the *z* axis, i.e., perpendicular to the screen—is frequently mapped to *x,y* touch input [33,50], sacrificing the relation between input and control task structures [19].

In this paper, we present HoloFlex, a glasses-free 3D smartphone featuring a flexible thin-film light-field display with thousands of microscopic lenses. 3D images are rendered on a high resolution Polymer Organic Light Emitting Diode (P-OLED) display by a ray-tracing algorithm that simulates a hexagonal array of 160×104 pinhole cameras distributed on a 2D plane. Each camera independently renders a wide-angle 2D image from a given position onto a circular 12-pixel wide block (see Figure 1). An array of 16,640 half-dome microlenses distributes the rays of each pixel block horizontally and vertically back into the eyes of the user, reconstructing the 3D scene (see Figure 2). In addition to multi-touch *x,y* and inertial input technologies, HoloFlex features bend sensors for interacting with the *z* dimension. This aligns with recent research on flexible devices suggesting that bend input is highly suited for interactions with the *z* dimension [6,27].

Contributions
This paper has three contributions. First, it presents the HoloFlex flexible light-field smartphone prototype with associated interaction techniques: glasses-free motion parallax and stereoscopy display, as well as the ability to deform the display for Z-Input. HoloFlex's unique features make it suitable for mobile interaction with 3D content.

UIST 2016, October 16-19, 2016, Tokyo, Japan
 ISBN 978-1-4503-4189-9/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984524

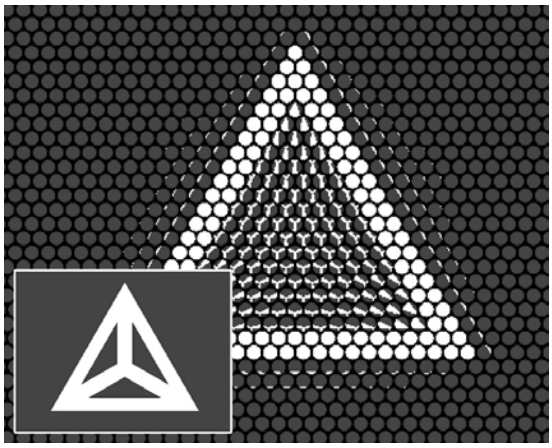

Figure 1. 3D Light-field rendering of a tetrahedron (inset bottom-left shows 2D rendition). Note the 12-pixel wide circular blocks rendering views from different camera angles.

Second, we present a preliminary study in which we compare the use of bend gestures vs. the use of a touch slider for *Z*-input in a 3D docking task. Results suggest that bend input performs significantly faster than a touch slider in this *Z*-input task, with similar accuracy. Third, we present three application scenarios that describe the use cases of the HoloFlex prototype: *a)* Light-field mobile editing of 3D models; *b)* Light-field mobile physical gaming; and *c)* Multiview light-field mobile videoconferencing.

RELATED WORK

Multi-view displays

The combined presence of stereoscopy—separate views for each of the user's eyes—and motion parallax—multiple view zones for different positions of the user's head—are critical in the perception of depth [22]. Several technologies have been used in the development *multi-view displays* capable of preserving depth cues:

Head-tracking and head-mounted displays

Multi-view capabilities can be simulated by head-tracking, as demonstrated by Benko et al. [4] in a tabletop-based prototype. Their system also allowed the user to interact with 3D objects, albeit by sacrificing mobility. Other solutions use virtual and augmented reality headsets such as the Oculus Rift [41] and Microsoft's Hololens [38]. While these devices are capable of providing convincing and immersive experiences of 3D environments, they pose problems for scenarios that require face-to-face interaction as they tend to obscure the user's face. Hao Li et al. [31] attempted to solve this problem, but their system added more bulk and required calibration before use.

Light-field Displays

Light-field displays, which have been around since the early 20th century, provide a natural way to visualize 3D scenes by emulating the field-like propagation of light. The two dominant technologies for generating light-fields are parallax barriers [19] and microlens arrays [32].

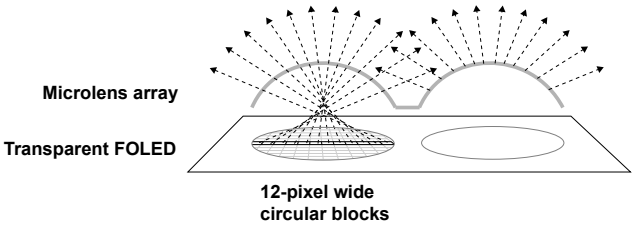

Figure 2. HoloFlex side close-up with 12-pixel wide pixel blocks and half-dome microlens array dispersing light rays.

Parallax Barriers

Some techniques for creating light-fields include directional backlighting (Fattal et al. [8]) and parallax barriers. Portallax [36] was a retrofitting 3D accessory for smartphones that featured a parallax barrier in combination with face tracking. Parallax barriers have also been used commercially in the HTC Evo 3D [17] and the Nintendo 3DS [40], albeit only to support stereoscopy.

Microlens Arrays

Some more recent implementations use microlens arrays to create light-field displays. Hirsch et al. [15] used a 2D display to image captured light-field content. They used LCDs with integrated optical sensors co-located at each pixel to record multi-view imagery in real-time. Hirsch et al. [14] implemented a display that reacts to incident light sources. Users can use light sources as input controls for applications such as the relighting of virtual 3D objects. Tompkin et al. [57] used a light pen captured through a light-field display, allowing the system to act as both input and output device. None of these systems are particularly portable.

Projector-Array Based Systems

An alternative way to displaying light-fields is to employ arrays of projectors. The HoloLeap project [1] used such a display coupled with the Leap Motion controller to provide 7-DOF object manipulation. On the other hand, Peng et al. [44] combined a custom projector-based system with a Kinect for 3D capture for light-field teleconferencing.

Interaction with 3D objects

Interacting with objects in virtual 3D worlds is a non-trivial task that requires matching physical controllers to translation and rotation of virtual objects, a problem studied extensively by Zhai [59]. This implies the coordination of control groups—translation, rotation—over several DOFs [37]. Balakrishnan et al. [3] presented a 4 DOF input device for 3D graphics manipulations. It resembles a mouse with a rounded bottom, allowing it to be tilted about two axes, providing it with two more degrees of freedom than a standard mouse. Froehlich et al. [8] presented two 6 DOFs desktop input devices that feature a custom three degree-of-freedom trackball used as a rotation sensor. Translation was separately controlled by applying force to the trackball in one device and by measuring the movement of the base in the other. Valkov et al. [58] used imperceptible shifts of objects along the Z-axis to minimize contradictions in visual depth cues as the user approaches the object in the

display. Hachet et al. [13] combined 2D interaction techniques with 3D imagery in a single interaction space. Reisman et al. [45] extended well-known rotate-scale-translate metaphors for 2D manipulation (such as pinch to zoom) into 3D. The authors proposed three or more finger interaction techniques that seem to provide direct manipulation of 3D objects in a multi-touch environment. Their techniques are, however, not appropriate for mobile devices, as they require bimanual multi-finger interactions. Other methods of 6-DOF manipulation [33,50] free up the second hand for tasks such as holding the mobile device, but require an indirect mapping of Z-axis manipulation to multiple finger touches in the *x,y* plane. This indirect mapping does not take into account the findings of Jacob et al. [19], which showed that performance and satisfaction improve when the structure of the task matches the structure of the input control. Yet another possibility is that of hand-pose reconstruction [42,52,55]. Kim et al. [25] demonstrated a wrist-worn system capable of reconstructing hand position enabling 3D interactions on a mobile device. One of the drawbacks of this approach is a lack of tactile feedback.

Flexible Phones and Bend Interactions

Flexible and shape-changing interfaces have enjoyed a lot of interest recently and date back to earlier work with projected shape-changing displays [23,29,53]. Lahey et al. [27] introduced PaperPhone, one of the first flexible smartphones. It featured a thin-film electrophoretic screen with a flexible circuit board capable of sensing screen deformations [51]. In PaperPhone, such bend gestures were used to navigate through items on a 2D display. Evaluations show that bend gestures are particularly suited for continuous single variable input [2,24]. They also appear well suited for Z-Input [6]. Lahey et al. [27] found that bend gestures that take directional cues into account feel more natural to users. Other work such as PaperFold [10], FlexSense [46] and FlexCase [47] evaluated the benefits of fold and bend gestures in handheld devices. A study conducted by Ahmaniemi et al. [2] found that bend gestures suited zooming or scaling actions at least as well as the more familiar pinch gesture. They observed that bend input reaches its maximum potential when controlling a one-dimensional continuous bipolar parameter. Burstyn et al. [6] investigated the combination of deformation and touch input on a handheld device. They found that the efficiency of interaction in pan and zoom tasks was improved over multi-touch interaction. We conclude that bend input has the potential to provide a parallel input dimension for *z* translation in addition to multi-touch for *x,y* translation.

DESIGN RATIONALE

In the design of our HoloFlex prototype we took the following design parameters into consideration:

Motion-Parallax and Stereoscopy

Stereoscopic displays—such as the one in the Nintendo 3DS [40]—typically provide binocular parallax but not motion parallax. Single-user motion parallax is attainable via methods such as head tracking [42]. However, this usually presents a perceptible delay, works only for a single user and requires additional hardware. While other fully immersive 3D displays exist, such as HoloLens [38] and Oculus Rift [41], these require the use of additional head-worn apparatus. In our design, we wanted to preserve both motion parallax and stereoscopy to make it easier for users to interact with 3D objects, for example, in 3D design tasks. We also wanted to make the device self-contained, and keep mobile sensors—such as the IMU—available for user input.

Natural 3D Glasses-free Display

We chose a display type that can provide the full range of depth cues with no additional hardware: a light-field display [29]. When presenting a 3D scene, a light field renders correct perspective to a multitude of viewing angles. When a user wants to observe the side of an object in a 3D scene, she simply moves her head as she would to view the side of a real world object, making use of her natural behavior and previous experiences. This means no tracking or training is necessary. And as multiple angles are naturally supported, the light-field display supports multiple simultaneous users.

Display Resolution, Dot Pitch and Microlens Array

One of the issues with 3D displays in general and light-field displays in particular is their limited resolution. In parallax barrier displays this is further compounded by the sparse spacing of the viewports. We used a 403 DPI retina display to allow for sufficient resolution to display a large number of pixel blocks while maintaining a small dot pitch per pixel block. We manufactured a microlens array of 16,640 microscopic plastic lenslets, each with a radius of 375 μm, allowing for a sufficiently small dot pitch per pixel block to see a fused 3D image at normal distances from the screen.

Mobile Raytracing

While one could capture images suitable for a light-field display using an array of cameras or a light-field camera [29], more typically our content is generated as 3D graphics. This requires ray tracing, which is computationally expensive. However, limiting the 3D scene to simple low polygon count models allows for real-time rendering and 3D interactive animations.

Squeeze for Z-Input

3D translation on mobile platforms can be difficult to perform with a single *x,y* touch input. Since the third axis is perpendicular to the touch input plane, no obvious control

Figure 3. HoloFlex light-field smartphone prototype with flexible microlens array.

of Z-input is available via touch. Indeed, current interaction techniques in this context involve the use of indirect intermediary two-dimensional gestures [55,7,35]. While tools exist for bimanual input, such as a thumb slider for performing *Z* operations (referred to as a Z-Slider), these tend to obscure parts of the display space. Instead, we propose a bimanual combination of dragging and bending to control 3D translation. Here, bend input is performed with the non-dominant hand holding the phone, providing an extra input modality that operates in parallel to *x,y* touch input by the dominant hand [12]. The gesture we chose for bend input was *squeezing* as defined by Burstyn et al.: "*squeezing* involves gripping the display in one hand and applying pressure on both sides to create concave or convex curvatures" [6]. Since the user's finger follows the direction of the object's movement *squeezing* provides direct and integral control over 3D translation. This is fundamentally different from 3D pressure input as it provides valuable force and kinesthetic feedback.

Flexible Display

Aside from providing HoloFlex with the capacity for bend input, our choice of a flexible display form factor provided other benefits. A flexible display is ideally suited for working with 3D objects because it can be molded around the 3D design space to provide up to 180 degree views of an object. Since the *z* range of many light-field displays is limited a flexible light-field display is particularly suited for say cylindrical objects. Furthermore, bending the display along the *z* axis also provides users with passive haptic force feedback about the *z* location of the manipulated 3D object.

IMPLEMENTATION

Figure 3 shows the prototype HoloFlex smartphone. The phone consists of five layers, as explained in Figure 4: 1) A microlens array, 2) A flexible touch input layer, 3) A high resolution flexible P-OLED, 4) Bend sensor and 5) Rigid electronics and battery.

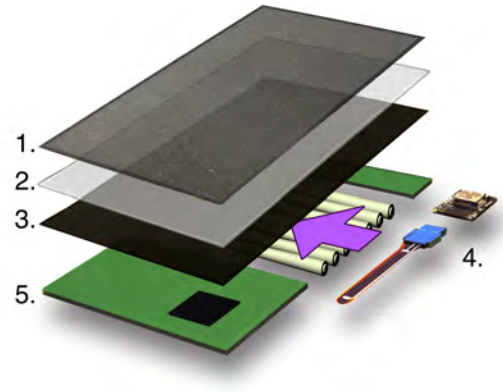

Figure 4. Layers of the HoloFlex prototype: 1. Microlens array, 2. Flexible touch input layer, 3. High-resolution flexible P-OLED, 4. Bend sensor (shown besides the device for clarity), 5. Rigid circuit and flexible battery.

1. Microlens Array

The microlens layer is a 3D printed flexible plastic lens array of our own design consisting of 16,640 half-dome shaped lenslet. The lenslets are printed on a flexible optically clear substrate 500 µm in thickness, laid out in a 160 × 104 hexagonal matrix with a distance between lenslet centres of 750 µm. Each lenslet is a sphere with a radius of 375 µm submerged in a substrate, with its top protruding 175 µm. A optically opaque mask printed onto the substrate fills the space between the lenslets, separating the light-field pixel blocks from one another and limiting the bleed from unused pixels. We designed the lenslet specifications using OpticalRayTracer 9.2. We sent our 3D designs of the microlens array along with the 2D optical mask to 3D optics printing company [34] for manufacturing. In order to minimize Moiré effects, we chose the spacing between the microlenses in a way that does not align with the underlying pixel grid, rotating the microlens array 0.47 degrees, as recommended by Ji et al. [21]. The microlens layer was glued to the display, after which the alignment was fine-tuned in software by visual inspection of calibration patterns consisting of white circles centered at each lens.

2. Touch Input Layer

The touch input layer consists of a flexible capacitive touch film by LG Display that senses *x,y* touch with a resolution of 1920 × 1080 pixels.

3. Display Layer

The display layer consists of a 403 dpi 121 × 68 mm LG Display P-OLED display with a display resolution of 1920 × 1080 pixels.

4. Bend Sensing Layer

HoloFlex features one bidirectional 2" FlexPoint bend sensor placed horizontally in the center of the display. It senses horizontal bends when the phone is held in a landscape orientation. Bend sensor values are sampled and communicated over a Bluetooth radio to the Android board by an RFduino [46] module running RFduino 2.3.1.

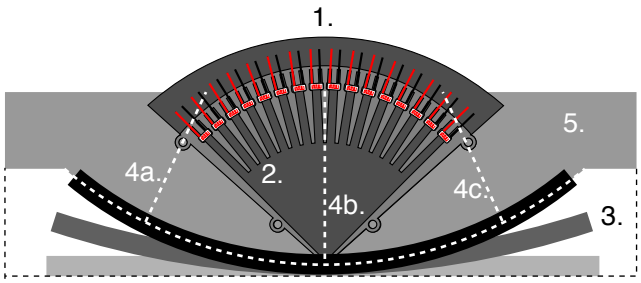

Figure 5. Light angle measurement tool: 1. Sixteen photoresistors. 2. Light-proof case. 3. Measurements can be obtained at different bend radii. 4abc. Measurements can also be obtained at different locations on the display. 5. Calibration support for the measurement tool.

5. Rigid Circuitry and Battery Layer

The final layer consists of a 66 × 50 mm Android circuit board with a 1.5 GHz Qualcomm Snapdragon 810 processor and 2 GB of memory. The board runs Android 5.1 and features an Adreno 430 GPU supporting OpenGL 3.1. The circuit board was placed such that it forms a rigid handle on the left back of the HoloFlex device. The handle allows users to comfortably squeeze the device without breaking the screen. A 1400 mAh flexible array of 7 pile batteries of our own design was placed in the center back of the device such that it can deform with the screen.

Rendering Algorithm

The microlens array redistributes the light emanating from the P-OLED pixels into individual directions, allowing the HoloFlex unit to modulate the light output sent to each viewing direction. Each pixel block rendered on the light-field display consists of an ~80 pixel circular image of the scene from a particular virtual camera position on the x, y plane (see Figure 1). This approximates a 2D array of pinhole cameras with a 59 degree field of view and resolution of 80 pixels. To calculate the direction of the ray that corresponds to every specific pixel, we first determine the position (x, y) of that pixel with respect to its closest lenslet. The direction of the ray $\vec{v}$ at that pixel is then obtained by applying two rotations to the surface normal $\vec{n}$:

$$\vec{v} = R_{\vec{n}}(\varphi)\, R_y(\theta)\, \vec{n}$$

where R_y and $R_{\vec{n}}$ are the counter-clockwise rotations about the y-axis and $\vec{n}$ respectively, and:

$$\theta = \|(x, y)\| \cdot K_{lens},$$
$$\varphi = \tan^{-1}(y, x),$$

$K_{lens} = 1.365$ rad/mm for our prototype, determined empirically as explained below. The scene is rendered using a ray-tracing algorithm running on the GPU of the phone. Since we needed to render objects that appear to float above the display, our cameras also render what is behind them. We implemented custom OpenGL shaders in GLSL ES 3.0 for real-time light-field rendering. The resulting scene and input data—touch and bend—are managed by a Unity 5.1.2 application running on the phone.

Compensating for Display Bend

Bending HoloFlex causes an unintended distortion of the displayed rendering. We compensate for this effect in software, as follows. Our model assumes the curvature of the display to be approximately cylindrical when bent. We estimate the cylinder radius on the basis of bend sensor readings. A linear regression on these readings provides a curvature function — bend sensor value b vs. the inverse curvature radius r^{-1}. As mentioned, during ray tracing the direction of each ray is calculated on the basis of the surface normal $\vec{n}$. Thus, having an estimate of the curvature radius r for pixels corresponding to a lens at the position $\vec{l} = (x_l, y_l)$, we adjust the origin of the ray $\vec{o}$ to be:

$$\vec{o} = \left(x_l \cdot \mathrm{sinc}\left(\frac{x_l}{r}\right), y_l, \mathrm{x}_l \cdot \mathrm{sinc}\left(\frac{x_l}{r}\right) \tan\left(\frac{x_l}{2r}\right)\right),$$

with the surface normal $\vec{n}$ set to:

$$\vec{n} = \left(-\sin\left(\frac{x_l}{r}\right), 0, \cos\left(\frac{x_l}{r}\right)\right),$$

where $\mathrm{sinc}(\theta) = \frac{\sin\theta}{\theta}$, which we approximate using a finite product from the identity:

$$\mathrm{sinc}(\theta) = \prod_{k=1}^{\infty}\left[1 - \frac{4}{3}\sin^2\left(\frac{\theta}{3^k}\right)\right].$$

The formulae above are designed to avoid underflow or overflow errors even when the curvature radius r is large, (i.e., the display is flat,) by using r^{-1} directly.

Compensating for Lens Shift

Bending the device also causes a slight shift of the microlens layer versus the underlying display layer. More precisely, the lenslets shift outwards in relation to the pixels in the x dimension. This is a problem as knowing the centers of the lenslets is critical for accurate raytracing and light-field generation. We correct for this by enlarging the pixel block spacing associated with the lenslet along the x-axis as bend increases. The pixel block of size $\vec{w}$ (used in nearest lenslet calculation) is updated to:

$$\vec{w} = (x_w + K_{bend} \cdot r^{-1}, y_w)$$

where (x_w, y_w) = (750µm, 649.5 µm) is the original (unbent) pixel block spacing, r is the curvature radius as per the above, and $K_{bend} = 0.44$ is a constant obtained experimentally, as follows.

Empirical Determinants

To empirically determine the constants K_{lens} and K_{bend} used in the previous section, we created a light angle measurement tool (see Figure 5). It consists of 16 photoresistors spaced radially every 6 degrees around a central light entry point and otherwise enclosed in a black light-proof case. The voltages between the terminals of the photoresistors were sampled using an Arduino Mega, in

turn connected to a computer for analysis. We also designed a set of support structures to bend the HoloFlex display to known radii and to position the measurement tool perpendicular to the surface of the display at different points along its x-axis. The axis of measurement was aligned with the x-axis of the display (lengthwise) at all times.

Pixel Location and Light Output Angle

We first measured the effect of the shift of a pixel relative to a lenslet center on the angle of light emanating from a flat display. We rendered a 2 × 4 pixel monochromatic (green) rectangle under the center of each lenslet. We recorded light output with our measuring tool for 500 ms and moved the rectangle in the x direction by ¼ of a pixel[1]. This process was repeated until the rectangle was behind the center of a lenslet two positions away. We recorded the x shift at which each photoresistor value peaked, and performed a linear regression to the locations of these peaks. The slope of the regression determined K_{lens}, which was 1.365 radians/mm with $R^2 = 0.999$. Since the pixel size is 0.063 mm, the angle of separation of the light output of neighboring pixels is 4.9 degrees, with a total view angle of 58.7 degrees.

Effects of Bend on the Angle of Light Output

As mentioned, bending the display shifts the lenslet array relative to the pixels. To obtain the constant K_{bend} used for correcting this effect, the device was curved to known radii. Again, we used a 2 × 4 pixel monochromatic (green) rectangle rendered under the center of each lenslet. We used 6 support structures with radii 113.32 mm, 136.03 mm, 175.53 mm, 257.05 mm, 506.60 mm and ∞ (flat). For each curvature, we measured the light output at three positions along the x-axis: -50 mm, 0, and 50 mm (see Figure 5, 4abc). We manually adjusted the x-offset of the rendered rectangle to match the original light output values. We recorded the resulting offsets and fit linear regressions per curvature to link the shift of the microlens array to locations along the surface. The resulting linear regression had a slope of 0.58 with R^2 = 0.99 and a y-intercept of 0. This means that when r = 100 mm, the microlens array "stretches" relative to the display in the x dimension by a factor of 1.0058. Since the spacing between the centers of the lenslets along the x-axis is 0.750 mm, we determined $K_{bend} = 0.44$.

PRELIMINARY USER STUDY

We evaluated the performance of bend input in a flexible 3D smartphone as a method for z-input: we conducted an experiment that compared the efficiency of bend input with that of using a touch slider (the current norm) for z translations of a 3D object on the display. Both conditions used touch input for x,y translations.

[1] Subpixel movement is possible since our lenslets are not located at integer multiples of the pixel positions.

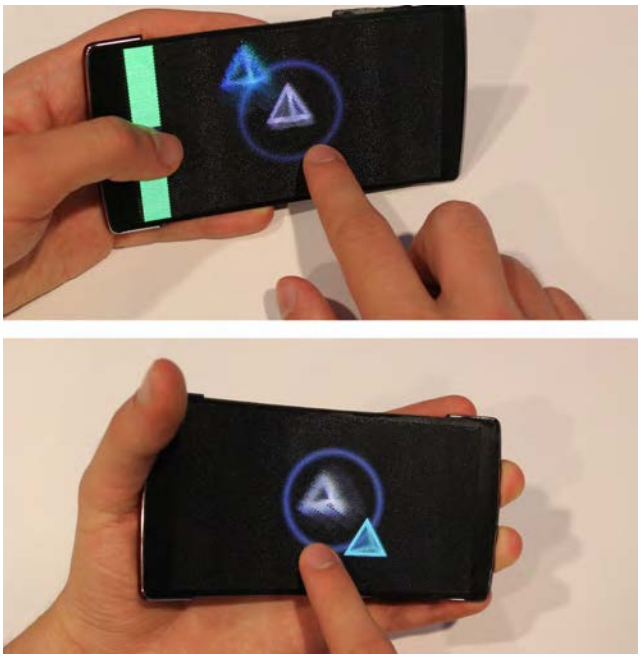

Figure 6. Typical tetrahedral cursor and target position during the experiment. Z-slider condition (top) and Bend condition (bottom).

Task

The task was based on a docking experiment designed by Zhai [59]. Subjects were asked to touch a 3D tetrahedron-shaped cursor, always placed in the center of the screen, and align it in 3 dimensions to the position of a 3D target object of the same size and shape. Figure 6 shows a 3D rendition of a sample cursor and target, a regular tetrahedron with edge length of 17 mm, as used in the experiment.

Trials and Target Positions

For every trial, the 3D target was randomly placed in one of 4 x,y positions equidistant from the center of the screen, and 4 positions distributed along the z axis, yielding 16 possible target positions. Each target position was repeated 3 times yielding a total of 64 measures per trial.

Experiment Design

The factor was *Z-Input Method,* with two conditions*:* Bend gestures vs. the use of a Z-slider. In both conditions, participant experienced the light-field with full motion parallax and stereoscopy. The display was held in landscape position by the non-dominant hand, and the cursor was operated by the index finger of the dominant hand. In the Z-Slider condition, users performed z translations of the cursor using a Z-slider on the left side of the display (see Figure 6 top), operated by the thumb of the non-dominant hand (Figure 6 bottom). In the Bend condition, users performed z translations of the cursor via a squeeze gesture performed using their non-dominant hand. We used a within-subject repeated measures design fully counterbalancing experimental conditions. The experiment

was controlled by a C# script running on a Windows 8 PC, which communicated with Unity 3D software on the phone via a Wi-Fi network.

Dependent Variables

Measures included time to complete task (*Movement time*), and distance to target upon docking in the *x, y* and *z* dimensions. *Movement time* measurements started when the participant touched the cursor, until the participant released the touchscreen. *Distance to target* was measured as the mean Euclidian distance between the 3D cursor and 3D target locations upon release of the touchscreen by the participant. All measures were communicated to the C# script on the Windows PC over a Wi-Fi network after completion of each trial and written to a file for off-line analysis.

Hypothesis

H1. *Bend input significantly improves movement time to target in performing a 3D docking task over a Z-Slider.*

H2. *Bend input does not differ significantly in accuracy (distance to target) from a Z-Slider in performing a 3D docking task.*

Questionnaires

After the experiment, users were administered a NASA TLX [39] questionnaire that used a 5 point scale to score Mental Demand, Physical Demand, Temporal Demand, Performance, Effort and Frustration.

Participants and Training

We invited 12 paid volunteers (10 male, 2 female, all right-handed) to participate in the experiment. All participants were graduate or undergraduate students with experience using touch input on handheld devices. Participants were required to have 20/20 vision and acceptable stereo vision, which was tested prior to admission to the experiment based on a method by Boritz et al. [5]. The test showed 3 tetrahedrons with the same geometry and size as the cursor and target in the experiment. Two of these tetrahedrons were located at the same *z* location. Participants were asked to tell which tetrahedron was different in depth in each trial. We conducted 15 trials with random depth differences. A participant was accepted only when she produced no more than 1 incorrect answer out of these 15 trials. Participants were trained before study trials by having them perform the experimental task using random targets until they achieved less than 10% improvement in docking time between trials.

RESULTS

We analyzed the dependent variables, averaged across multiple trials, using paired t-tests evaluated at an alpha level of 0.05. Questionnaires were analyzed using a Wilcoxon Signed Rank non-parametric test.

Mean Movement Time

Table 1 shows the mean movement times for the *Z-Input Method* conditions. Results show that mean *Movement time* was significantly lower (24%) in the Bend condition than in the Z-Slider condition ($t(11) = -1.898$, $p=0.042$, one-tailed).

Z-Input Method	Movement Time (s)	Distance to Target (mm)
Bend	**4.3* *(0.4)***	4.7 *(0.5)*
Z-Slider	**5.3* *(0.6)***	3.9 *(0.5)*

Table 1. Means and standard errors (*s.e.*) for movement times (in seconds) and distance to target (in mm) for each Z-Input Method. Significant differences indicated in bold with *.

Distance to Target

Table 1 also shows the mean accuracy for the *Z-Input Method* conditions. Results show that the mean *Distance to target* was not significantly different between conditions ($t(11) = 1.251$, $p=0.236$, two-tailed).

Questionnaire

We analyzed the NASA TLX questionnaires using a two-tailed Wilcoxon Signed Rank test. For *Z-Input Method*, participants indicated that Bend input (mean: 3.3; std. dev.: 0.7) was significantly more physically demanding than the Z-Slider (mean: 2.3 std. dev.: 1.2) ($Z=-1.981$, $p=0.48$). Other questionnaire items were not significant between conditions.

DISCUSSION

The results confirmed both hypothesis H1 and H2. While Bend input significantly improved *Movement time* in the docking task over Z-Slider input, there was no significant difference between input techniques in terms of accuracy. While we cannot conclude that accuracies were the same, it is unlikely that *Movement time* results for bend input were simply due to participants performing less accurately. Based on participants' comments, Bend input provided a more natural mapping than the Z-Slider. Bend input maps to the *z* dimension directly: as participants squeeze the display downwards, the cursor moves in the same direction. This is not the case for Z-Slider input, which moves orthogonally to the cursor. A second explanation for this result could be that less movement was required for Bend input than for Z-Slider input. That said, Bend input involved moving the muscles of the entire hand, while Z-Slider only involved the muscles of the thumb. Herein lies another possible explanation: that the thumb may be slower than a squeeze of the hand [60]. Finally, bend input may have provided a more integral approach to moving graphics simultaneously in all 3 dimensions over the Z-Slider.

According to qualitative observations, participants found Bend input significantly more physically demanding than the Z-Slider. In fact, participants reported Bend input induced fatigue. We believe this was because the current prototype is too rigid and too large to comfortably fit all hands. Based on participant comments, we would expect that an increased flexibility and smaller size of our prototype would improve our design.

Figure 7. Light-field physical gaming application.

APPLICATION SCENARIOS

We developed a number of applications that implement the use of bend input with a light-field flexible smartphone. All applications were developed in Unity 3D.

Light-field Editing of a 3D Print Model

Our first application demonstrates the use of bend gestures for Z-Input to facilitate the editing of 3D models, for example, for 3D printing tasks. Here, *x,y* positioning with the touch screen is used for moving elements of 3D models around the 2D space. Exerting pressure in the middle of the screen, by squeezing the screen with the non-dominant hand, moves the selected element in the *z* dimension. IMU data facilitates the orientation of 3D elements. This is enabled only when a finger is touching the touchscreen in order to avoid spurious rotational input. By bending the display into a concave shape, multiple users can examine a 3D model simultaneously from different points of view.

Light-field Physical Gaming

Our second application is a light-field game (Figure 7). The bend sensors and IMU in HoloFlex allow for the device to sense its orientation *and* shape. This provides gaming experiences that are truly imbued with physics: 3D game elements are presented on the light field display, and by means of elastic deformation, HoloFlex can be used as a physical, passive haptic input device. To demonstrate this, we developed a simplified version of the Angry Birds game [49]. Rather than using touch input, users can bend the side of the display to pull the elastic rubber band that propels the bird. This provides the user with passive haptic feedback representing the tension of the rubber band in the slingshot. To release the bird, the user releases the side of the display. The velocity with which this occurs is sensed by the bend sensor and conveyed to a physics engine in the gaming app, sending the bird across the display with the corresponding velocity.

Multiview Light-field Videoconferencing

Our third application is a 3D light-field video conferencing system. Augmenting HoloFlex with 3D depth camera(s) such as Project Tango [11], or a transparent flexible light-field image sensor [18], allows it to capture 3D models of real world objects and people. This allows HoloFlex to convey light-field video images viewable from any angle. As a proof of this concept, we sent RGB and depth images from a Kinect 2.0 capturing a remote user over a network as uncompressed video images. We used these images to compute a real-time colored point cloud in Unity3D. We then rendered this point cloud for display on the HoloFlex prototype (see Figure 8). Users can look around the 3D video of the remote user by bending the screen into a concave shape, while rotating the device with their wrist. This presents multiple local users with different viewpoints around the 3D video in stereoscopy and with motion parallax.

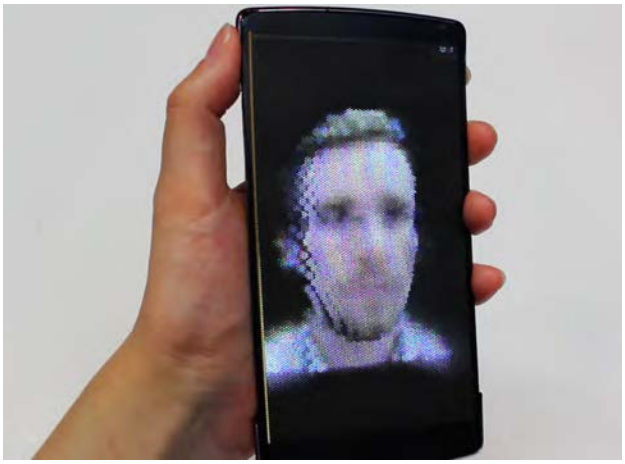

Figure 8. Light-field videoconferencing application.

LIMITATIONS

While we believe our results would generalize to other bendable smartphones with light-field displays, we should note that they are limited by the specific rigidity and size of our prototype. More flexible or smaller versions might produce different results. In addition, further increases in the resolution and angular fidelity of the light-field display might improve depth perception. Nevertheless, bending the display outwards will necessarily decrease the spatio-angular resolution of such a display. Evaluating the effects of these changes will be the subject of future work.

Hardware Limitations

One of the main drawbacks of the current light-field hardware is its low resolution as compared to, e.g., the use of a retina display with time-multiplexed stereo glasses. Nevertheless, this limitation was not crucial for our study. We predict resolution will be improved through the development of higher resolution displays. A second downside is the computational demand of the rendering algorithm. Although we expect the graphics capability of mobile phone GPUs to improve significantly over time, one solution might be to perform image processing in the cloud, and providing individual HoloFlex devices with pre-rendered images over a low-latency network. The quality of 3D printed optics is a limiting factor and induced noticeable

crosstalk between viewports that further limited the *z* range of the display. We found that objects outside ± 20mm of the surface of the display were too blurry to view. Due to the construction of our prototype, it can only be bent to a cylindrical radius of 10-15cm. Beyond that, the display and microlens array layers tend to separate. Better assembly procedures that allow the lenslets to be placed closer to the P-OLED as well as the use of flexible circuit boards would make the design thinner and more flexible. This would also make the interaction using bend gestures easier.

CONCLUSIONS

We presented HoloFlex, a 3D light-field flexible smartphone. HoloFlex uses a light-field display to provide stereoscopy and motion parallax cues without the need for head tracking or glasses. In addition to featuring multi-touch input for *x,y* translations, HoloFlex allows the use of bend input for *z*-axis operations. We presented a preliminary user study in which we compared the use of bend input for *z* translations in a 3D docking task with that of a touch slider. Results suggest bend input significantly improves movement time over Z-Slider input in this task. Finally, we presented three applications featuring use of bend input on a light-field smartphone, including 3D modeling, bend-enabled gaming and 3D teleconferencing.

REFERENCES

1. Adihikarla, V. K., Wozniak, P., and Teather, R. Hololeap: Towards efficient 3d object manipulation on light field displays. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, ACM (2014), 158–158.
2. Ahmaniemi, T. T., Kildal, J., and Haveri, M. What is a device bend gesture really good for? In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM (2014), 3503–3512.
3. Balakrishnan, R., Baudel, T., Kurtenbach, G., and Fitzmaurice, G. The rockin'mouse: integral 3d manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ACM (1997), 311–318.
4. Benko, H., Jota, R., and Wilson, A. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM (2012), 199–208.
5. Boritz, J., and Booth, K. S. A study of interactive 6 dof docking in a computerised virtual environment. In *Virtual Reality Annual International Symposium*, 1998. Proceedings., IEEE 1998, IEEE (1998), 139–146.
6. Burstyn, J., Banerjee, A., and Vertegaal, R. Flexview: An evaluation of depth navigation on deformable mobile devices. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, ACM (2013), 193–200.
7. Daiber, F., Falk, E., and Krüger, A. Balloon selection revisited: multi-touch selection techniques for stereoscopic data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ACM (2012), 441–444.
8. Fattal, D., Peng, Z., Tran, T., Vo, S., Fiorentino, M., Brug, J., and Beausoleil, R. G. A multi-directional backlight for a wide-angle, glasses-free three-dimensional display. *Nature* 495, 7441 (2013), 348–351.
9. Froehlich, B., Hochstrate, J., Skuk, V., and Huckauf, A. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (2006), 191–199.
10. Gomes, A., and Vertegaal, R. Paperfold: evaluating shape changes for viewport transformations in foldable thin-film display devices. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ACM (2015), 153–160.
11. Google. Tango. https://get.google.com/tango/.
12. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior* 19, 4 (1987), 486–517.
13. Hachet, M., Bossavit, B., Cohé, A., and de la Rivière, J.-B. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM (2011), 587–592.
14. Hirsch, M., Izadi, S., Holtzman, H., and Raskar, R. 8d: interacting with a relightable glasses-free 3d display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 2209–2212.
15. Hirsch, M., Lanman, D., Holtzman, H., and Raskar, R. Bidi screen: a thin, depth-sensing lcd for 3d interaction using light fields. In *ACM Transactions on Graphics* (TOG), vol. 28, ACM (2009), 159.
16. Hirsch, M., Wetzstein, G., and Raskar, R. A compressive light field projection system. *ACM Transactions on Graphics* (TOG) 33, 4 (2014), 58.
17. HTC. Evo 3d. https://en.wikipedia.org/wiki/HTC_Evo_3Dwiki/HTC_Evo_3D.
18. ISORG. Isorg and plastic logic co-develop the world's first image sensor on plastic. http://www.isorg.fr/actu/4/isorg-and-plastic-logic-co-develop-the-world-s-first-image-sensor-on-plastic_149.htm.
19. Ives, F. E. Parallax stereogram and process of making same., Apr. 14 1903. US Patent 725,567.
20. Jacob, R. J., Sibert, L. E., McFarlane, D. C., and Mullen Jr, M. P. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction* (TOCHI) 1, 1 (1994), 3–26.
21. Ji, C.-C., Luo, C.-G., Deng, H., Li, D.-H., and Wang, Q.-H. Tilted elemental image array generation method for moiré-reduced computer generated integral imaging display. *Optics express* 21, 17 (2013), 19816–19824.

22. Johnston, E. B., Cumming, B. G., and Landy, M. S. Integration of stereopsis and motion shape cues. *Vision research* 34, 17 (1994), 2259–2275.
23. Khalilbeigi, M., Lissermann, R., Kleine, W., and Steimle, J. Foldme: interacting with double-sided foldable displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, ACM (2012), 33–40.
24. Kildal, J., Lucero, A., and Boberg, M. Twisting touch: combining deformation and touch as input within the same interaction cycle on handheld devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, ACM (2013), 237–246.
25. Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM (2012), 167–176.
26. Kim, S., Cao, X., Zhang, H., and Tan, D. Enabling concurrent dual views on common lcd screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 2175–2184.
27. Lahey, B., Girouard, A., Burleson, W., and Vertegaal, R. Paperphone: understanding the use of bend gestures in mobile devices with flexible electronic paper displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 1303–1312.
28. Lanman, D., and Luebke, D. Near-eye light field displays. *ACM Transactions on Graphics* (TOG) 32, 6 (2013), 220.
29. Lee, J. C., Hudson, S. E., and Tse, E. Foldable interactive displays. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM (2008), 287–290.
30. Levoy, M., and Hanrahan, P. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM (1996), 31–42.
31. Li, H., Trutoiu, L., Olszewski, K., Wei, L., Trutna, T., Hsieh, P.-L., Nicholls, A., and Ma, C. Facial performance sensing head-mounted display. *ACM Transactions on Graphics* (TOG) 34, 4 (2015), 47.
32. Lippmann, G. Epreuves reversibles. photographies integrals. *Comptes-Rendus Academie des Sciences* 146 (1908), 446–451.
33. Liu, J., Au, O. K.-C., Fu, H., and Tai, C.-L. Two-finger gestures for 6dof manipulation of 3d objects. In *Computer Graphics Forum*, vol. 31, Wiley Online Library (2012), 2047–2055.
34. Luxexcel. Luxexcel — 3d printed optics. https://www.luxexcel.com/.
35. Martinet, A., Casiez, G., and Grisoni, L. The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces* (3DUI), 2010 IEEE Symposium on, IEEE (2010), 115–118.
36. Martinez Plasencia, D., Karnik, A., Martinez Muñoz, J., and Subramanian, S. Portallax: bringing 3d displays capabilities to handhelds. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, ACM (2014), 145–154.
37. Masliah, M. R., and Milgram, P. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM (2000), 25–32.
38. Microsoft. Hololens. https://www.microsoft.com/microsoft-hololens/en-us.
39. NASA TLX. http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html.
40. Nintendo. 3ds. http://www.nintendo.com/3ds.
41. Oculus. Rift. https://www3.oculus.com/en-us/rift/.
42. Oikonomidis, I., Kyriazis, N., and Argyros, A. A. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, vol. 1 (2011), 3.
43. Ogniewski, J., and Ragnemalm, I. Autostereoscopy and motion parallax for mobile computer games using commercially available hardware. *International Journal of Computer Information Systems and Industrial Management Applications* 3 (2011), 480–488.
44. Peng, Y., Zhong, Q., Han, X., Yuan, L., Wang, R., Bao, H., Li, H., and Liu, X. 42.4 l: Late-news paper: Footprint of scalable 3d telecommunication: Using integral light field display and kinect-based capture. In *SID Symposium Digest of Technical Papers*, vol. 44, Wiley Online Library (2013), 589–592.
45. Reisman, J. L., Davidson, P. L., and Han, J. Y. A screen-space formulation for 2d and 3d direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ACM (2009), 69–78.
46. Rendl, C., Kim, D., Fanello, S., Parzer, P., Rhemann, C., Taylor, J., Zirkl, M., Scheipl, G., Rothländer, T., Haller, M., et al. Flexsense: a transparent self-sensing deformable surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM (2014), 129–138.
47. Rendl, C., Kim, D., Parzer, P., Fanello, S., Zirkl, M., Scheipl, G., Haller, M., and Izadi, S. Flexcase: Enhancing mobile interaction with a flexible sensing and display cover. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM (2016), 5138–5150.
48. RFDuino. http://www.rfduino.com/.
49. Rio. Angry birds. https://www.angrybirds.com/.
50. Scheurich, D., and Stuerzlinger, W. A one-handed multi-touch mating method for 3d rotations. In *CHI'13*

Extended Abstracts on Human Factors in Computing Systems, ACM (2013), 1623–1628.

51. Schwesig, C., Poupyrev, I., and Mori, E. Gummi: a bendable computer. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2004), 263–270.
52. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM (2015), 3633–3642.
53. Steimle, J., Jordt, A., and Maes, P. Flexpad: highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 237–246.
54. Techradar. Philips to launch glasses-free 3d tv in 2013. http://www.techradar.com/us/news/television/hdtv/philips-to-launch-glasses-free-3d-tv-in-2013-713951.
55. Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., and Pauly, M. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, vol. 34, Wiley Online Library (2015), 101–114.
56. Telkenaroglu, C., and Capin, T. Dual-finger 3d interaction techniques for mobile devices. *Personal and ubiquitous computing* 17, 7 (2013), 1551–1572.
57. Tompkin, J., Muff, S., Jakuschevskij, S., McCann, J., Kautz, J., Alexa, M., and Matusik, W. Interactive light field painting. In *ACM SIGGRAPH 2012 Emerging Technologies*, ACM (2012), 12.
58. Valkov, D., Giesler, A., and Hinrichs, K. H. Imperceptible depth shifts for touch interaction with stereoscopic objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2014), 227–236.
59. Zhai, S. *Human performance in six degree of freedom input control*. PhD thesis, University of Toronto, 1995.
60. Zhai, S., Milgram, P., and Buxton, W. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1996), 308–315.